

iTAP Switch Chip Engineering Specification

"Chiron Chip"

Appendix B
9/00

AUTHOR: M. Renault, D. Toebes, F. Carter
REVISION: Revision 0.30
DATE: May 12, 2000
APPROVED: _____

THIS DOCUMENT IS THE PROPERTY OF ONEX COMMUNICATIONS CORPORATION AND IS DELIVERED ON THE EXPRESS CONDITION THAT IT NOT BE DISCLOSED, REPRODUCED IN WHOLE OR IN PART, OR USED FOR MANUFACTURE FOR ANYONE OTHER THAN ONEX COMMUNICATIONS CORPORATION WITHOUT ITS WRITTEN CONSENT, AND THAT NO RIGHT IS GRANTED TO DISCLOSE OR SO USE ANY INFORMATION CONTAINED IN SAID DOCUMENT. THIS RESTRICTION DOES NOT LIMIT THE RIGHT TO USE

1 Overview

The iTAP Switch Element (iTSE) is a communications chip which can be used as a stand-alone device to implement a 12 x 12 port switching fabric. When combined with other iTSEs it is possible to create larger switch fabrics up to 1728 x 1728 ports for a 5-stage banyan network or up to 572 x 572 ports for a 5-stage Clos network.

Each port of an iTSE can simultaneously carry a mixed traffic load of TDM traffic, ATM traffic, and Packet traffic.

The iTAP switch fabric (comprised of one or more iTSEs) will typically be used as the interconnect scheme between iTAP Port Processors (ITPP).

Feature Highlights

- Synchronous Switching Architecture - All links which interconnect iTSE and ITPP ports are synchronized to a common data clock and row start reference.
- Bufferless Switching Fabric - Packet buffering is implemented via a combination of input and output queues within the ITPPs.
- For TDM traffic, the iTSE will support Time-Space-Time switching.
- The switching granularity for TDM traffic is at the VT1.5 level.
- For ATM/IP traffic, the iTSE will support a self-routed switching scheme. Since the iTSE will not implement packet buffers, a 2 phase switching algorithm is used. During phase 1 self-routed request messages will be transmitted across the switching fabric in a overlay control channel which matches the data interconnection paths. A "knockout" principle is then used to determine which requests will be serviced. The actual ATM/IP data is then sent through the switch fabric during phase 2. Requests not serviced during phase 1 will typically be re-requested during the next switch arbitration cycle.
- The switching granularity for ATM/IP traffic will be 64-byte fixed length PDUs.

1.1 Conventions in this Specification

The following conventions are used in this specification:

1.1.1 Terms and Concepts

Before proceeding to describe the operations of the iTSE, it will be useful to describe some terms and concepts which will be used throughout this document. The terms presented here are described in detail in Section 2, they are presented here only in a summary format.

Port

A port is physical interface on the iTSE which is used to interconnect to other iTSEs or ITPPs to form a switching fabric. The iTSE will have 12 input ports and 12 output ports. Each port is comprised of multiple LVDS channels. Specifically, an input port consists of 2 LVDS channel inputs and one LVDS channel output while an output port consists of 2 LVDS channel outputs and one LVDS channel input.

Link

This is the term used to describe the connection between the output port on one iTSE or ITPP and the input port on another iTSE or ITPP. The link physically consists of the circuit board wires to interconnect the LVDS channels. A link will always connect the bundle of LVDS channels from a single output port to a single input port, i.e., mixing of LVDS channels between links is not allowed. The LVDS channel bundle which makes up a link will consists of 3 LVDS pairs, 2 pair are used to carry the data traffic and 1 pair is used as an overlay network in the reverse path for carrying arbitration grants.

[LVDS] Channel

Individual LVDS channels will be bundled together to form a single link. The terms "LVDS channel" and "channel" will be used interchangeably in this document.

Row

The synchronous switching mechanism within the iTSE operates on "row" boundaries. The term row will be used to describe the traffic which is carried across a link during a single row

Proprietary and Confidential Information of Onex Communications Corporation

time. Row start times will occur at a 72 kHz rate (13.9 us). A row of data will consist of all the data carried on the link during a given row time. The row of data will be byte interleaved across multiple LVDS channels in order to achieve an aggregate data rate of 4.4 Gbps across the link.

Frame

A frame consists of a group of 9 rows which results in a frame rate of 8 kHz.

Group

The row structure is subdivided to carry 96 groups. Each group will be comprised of a block of 16 slots. The fixed length data PDUs are mapped into these groups (one PDU per group). The concept of switching a group of data (or a single PDU) is reserved for data carried in a single group.

Slot

In addition to subdividing row into groups, the row can also be subdivided into slots. A slot will be 36-bits wide. TDM traffic will be mapped onto the row using the slot terminology. Switching on a slot basis is reserved for TDM traffic.

PDU

Protocol Data Unit. A PDU will be defined for the iTAP chip which will be used to carry either ATM cells or IP Packets. Since the PDU will be fixed to a length of 64-bytes, longer IP packets will need to be fragmented and carried through the switch fabric on multiple PDUs.

Speedup

Concept where the switch fabric I/O ports each run faster than the external line rate. The ratio of switch fabric port speed / external line rate is the speedup. Speedup helps reduce input and output blocking through the switch.

Strictly Nonblocking

A switch is strictly nonblocking if a connection can always be set up between any idle input and output without the need to rearrange the paths taken by existing connections.

Recirculating Buffers

Used in a switch fabric. If multiple cells are destined to go through the same switch path, only one is allowed through and the rest are sent to the recirculating buffer where they will be looked at during the next switch cycle. This is frequently done to support multicasting. Also, recirculating buffers are sometimes timestamped so the cell stored in them will be discarded if it isn't forwarded within a given time interval. One thing to watch out for when using recirculating buffers is to prevent cell reordering, cells must be forwarded out the output port of the switch in the same order they're received at an input port.

1.1.2 Byte and Bit Ordering

Byte order is big-endian, bit ordering is little-endian (LSB is bit 0). This is shown below.

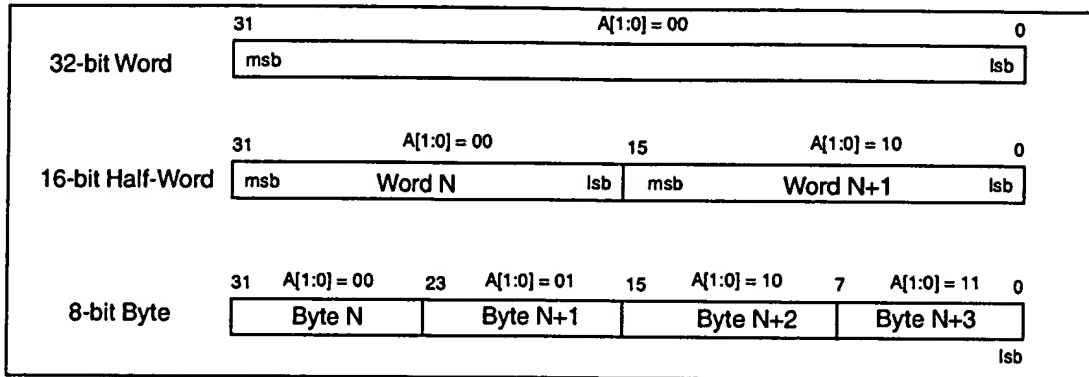


Figure 1-1: Byte and Bit Ordering Conventions for this Specification

1.2 References

Onex Communications Internal Documents -

- [1]
- [2]
- [3]
- [4]
- [5]
- [6]
- [7]
- [8]
- [9]
- [10]

Papers on Switching -

- [11] S. Liew, M. Ng, and C. Chan, "Blocking and Nonblocking Multirate Clos Switching Networks," *IEEE/ACM Trans. Networking*, vol. 6, no. 3, pp. 307-318, June 1998.

Nice simple review of Clos networks in section II.

- [12]
- [13]

Proprietary and Confidential Information of Onex Communications Corporation

1.3 Requirements

1. RESERVED

100-0000000000

2 iTAP Switch Element Functional Description

This section will provide the architectural overview of the iTSE. The objective here is to describe *what* the iTSE does, not how it is implemented. Some implementation concepts may be expressed in this section to aid in describing what the iTSE does, these implementation concepts may not reflect the actual implementation of the iTSE and are not constraints on the implementation approach to be chosen. The actual implementation will be provided in later sections of this document.

2.1 Switch External Interface

The figure below illustrates the iTAP Switch Element system I/O signals.

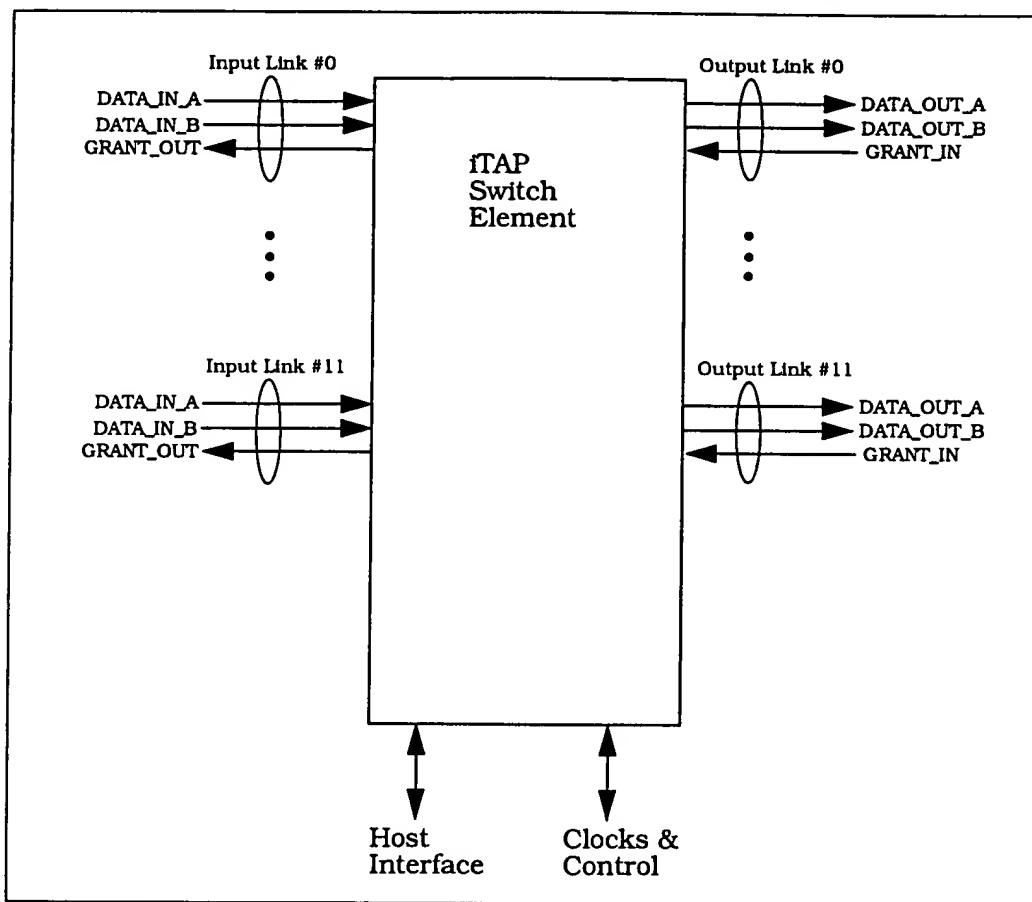


Figure 2-1: Switch Interface Block Diagram

Input Link's -

Each of the 12 Input Links is comprised of 3 high speed serial LVDS I/O signal pairs. The Input Links provide the data traffic input to the iTSE and the grants back to the previous stage.

The DATA_IN_A and DATA_IN_B pairs are used together to form a single high speed "logical" serial data input stream. This data input stream is used to carry both data traffic and the Request Elements which are used for bandwidth arbitration. The GRANT_OUT serial pair provides the output control channel for this Input Link.

All three LVDS pairs associated with an input link will always be connected as a group to the Output Link of the source iTSE or iTTP.

Multiple LVDS pairs must be used to create the single logical high speed Gbps serial stream because the maximum speeds currently supported by the candidate silicon vendors are less than what is need for an individual data link.

Output Link's -

The output link configuration matches the input link configuration.

Proprietary and Confidential Information of Onex Communications Corporation

2.2 Serial Link Formats

This section describes the formats for the serial links which are used to interconnect the iTSEs and iTPPs.

Since the output link of an iTSE would be directly connected to the input link for the next stage iTSE or an output Port Processor, these data structure describe the serial data formats for both the input and output links.

2.2.1 Design Objectives for Sizing Links

RESERVED.

2.2.2 Data Link

The 4.4 Gbps serial data stream is actually implemented using multiple lower speed serial streams. For this discussion, how the 4.4 Gbps serial data is split between the lower speed streams is not relevant, always view the serial data associated with an individual link as a single 4.4 Gbps stream.

As shown in the figure below, the serial data link channel is organized into slots, rows and frames. A "slot" consists of a 4-bit tag field and a 32-bit data (payload) field. The timing of rows and frames is architected to match that of Sonet/SDH frame timing. This will simplify the switching of TDM traffic which originates in Sonet/SDH payloads.

The last 20 slots are reserved for Link Overhead and may not be used to carry TDM or data traffic. The frame is transmitted from left to right (slot 0 to slot 1699) and top to bottom (row 1 to row 9). The msb of each slot is transmitted first. Note that for the data stream, a single PHY channel will not be capable of running at 4.4 Gbps, therefore the data stream will be split between two PHY channels each running at 2.2 Gbps. If the 1700 slot row is viewed instead as a 7560 byte row, the odd bytes will be transmitted on Phy channel A and the even bytes will be sent via channel B. The msb of each byte will be transmitted first. The row byte numbering start at 0, thus the last byte of the row is byte number 7559.

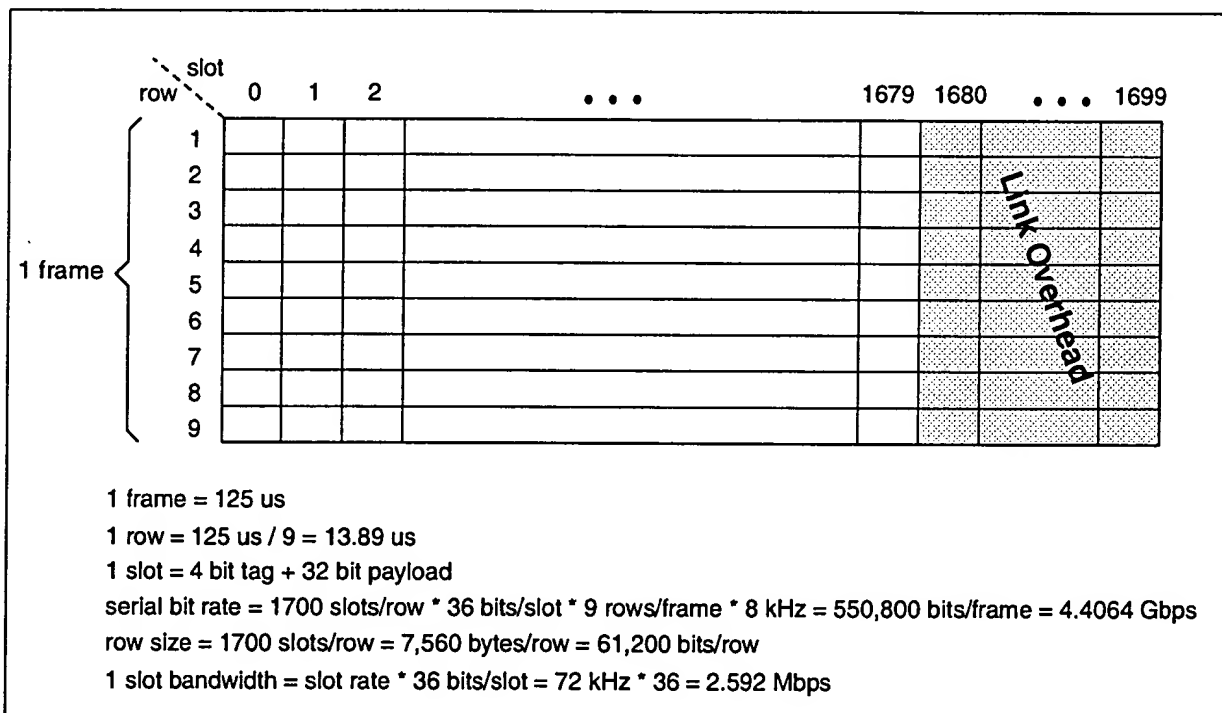


Figure 2-2: Data Frame Structure

For the transport of data PDUs, a block of 16 slots is required. The figure below illustrates one example of how the row slots may be allocated for carrying PDUs and Request Elements. As shown, the maximum PDU capacity for a row is 96. The term for block of 16 slots which is capable of carrying a single PDU is "group".

Note: Figure 2-3 illustrates just one partitioning of the row slots into groups and request elements, the implementation will allow flexibility in changing the row structure if necessary.

For each group in the row, 1.5 slots of bandwidth are required for carrying a 48-bit Request Element (RE). Figure 2-3 illustrates how 2 REs are inserted into 3 slots within each of the first 24 groups. All the REs need to be carried within the row as early as possible in order to allow the REs to ripple through the multi-stage switch fabric as soon as possible after the start of a row (see the Arbitration section for complete details). One option (not shown in this figure) would have been to send all 96 REs in the first 64 slots of the row. This is not being done because of the implementation

Proprietary and Confidential Information of Onex Communications Corporation

2.2.4 Data Traffic

RESERVED

[illegible]

Proprietary and Confidential Information of Onex Communications Corporation

2.3 iTAP System Implementation

RESERVED.

2.3.1 Clos Networks

RESERVED

Proprietary and Confidential Information of Onex Communications Corporation

2.4 ITSE Switching Examples

This section will provide examples which explain how data is sent through a 3-stage ITAP switch fabric. The examples will also summarize the control messages required to configure the switch connections.

2.4.1 Single PDU

The figure below illustrates a typical arbitration and data passing sequence for a data PDU.

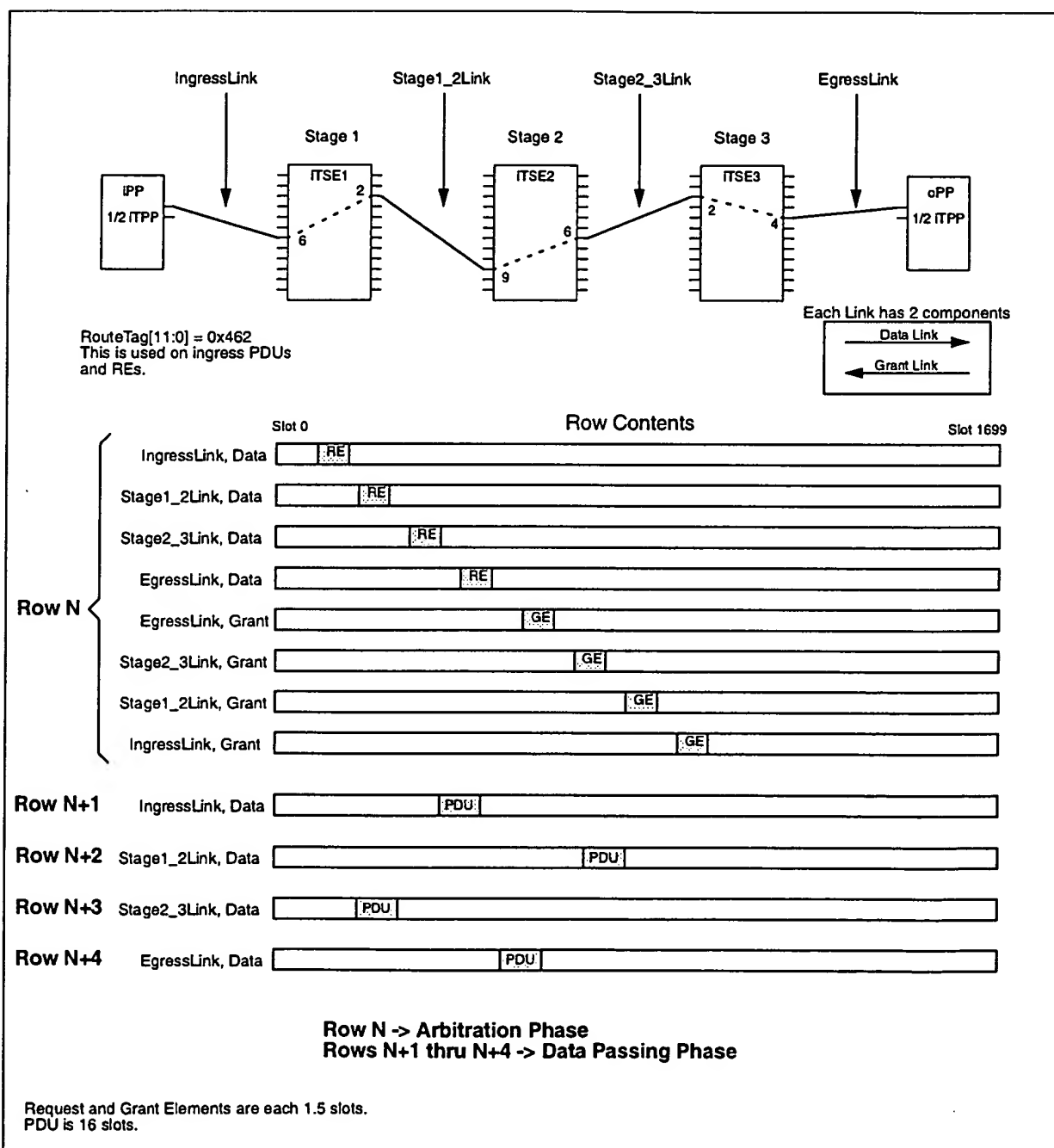


Figure 2-5: Single PDU Through a 3-Stage Switch Example

Proprietary and Confidential Information of Onex Communications Corporation

Control & Configuration -

In order to pass a data PDUs through the switch fabric, control messages is not used to set up the switch path. Instead, a "self-routed" concept is used. This means each RE, GE, and PDU sent through the switch fabric contains a RouteTag which identifies the path through the switch.

For each received ATM cell or IP packet, the iPP will classify which flow the data belongs to. The iPP will contain route tables which will contain the RouteTag field to be used to forward the data through the switch. When a new flow is established, only the PP route tables need to be updated, the switch doesn't require any updates.

1502_OVERVIEW.A11

Proprietary and Confidential Information of Onex Communications Corporation

2.4.2 Single TDM VT1.5 Channel

The figure below illustrates a typical TDM data flow through the a 3-stage switch fabric.

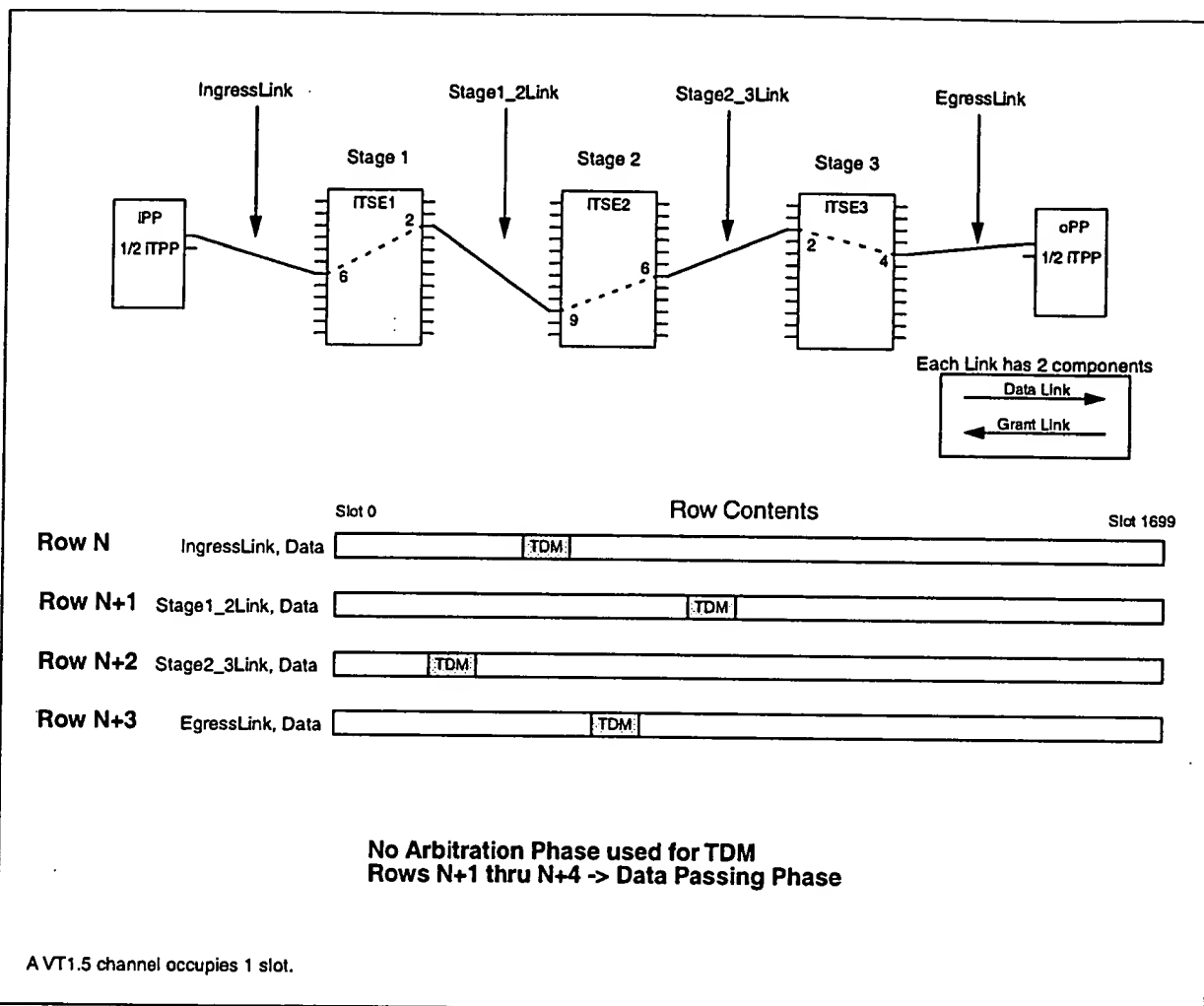


Figure 2-6: Single TDM VT1.5 Through a 3-Stage Switch Example

Control & Configuration -

- RESERVED

3 iTSE Implementation Overview

A block diagram of the iTSE is shown in Figure 3-1. This section will provide a brief overview of what each module in the block diagram does.

3.1 Datapath & Link BW Arbitration (per link modules)

This group of modules is instantiated 12 times, once for each I/O link the iTSE supports. These modules implement the core data path switching functions. A summary of the signals on the 3 internal datapath buses is shown in Section 3.4.

3.1.1 Data Stream Deserializer

The feature highlights of this module are:

- Synchronize to the incoming serial data stream and then reassemble the row stream which is transported using two physical Unilink channels. Provide FIFO'ing on each incoming serial stream so that the streams may be "deskewed" prior to row reassembly.
- Recover the 36-bit slot data from the row stream forward it a third FIFO which will be used for deskewing the 12 input links. This deskewing will allow all the input links to forward slot N to the switching core simultaneously. The link deskewing is controlled by the Link Synchronization & Timing Control module.
- Continuously monitor the delta between where slot 0 of the incoming row is versus the internal row boundary signal within the iTSE. This result will be reported to the Link RISC Processor and will be used as part of the ranging process to synchronize the iTPP connected to the input link (this would be a first stage function only).

The detailed description of this module is provided in Section 11.

3.1.2 Data Stream Demapper

This module is responsible for extracting the data from the incoming serial data links. The feature highlights of this module are:

- Demapping of the input link slots. This means based on the input slot number determine if the traffic is TDM, PDU, or a Request Element. The determination is based on the contents of the Demapper RAM.
- For TDM traffic, determine the destination link and row buffer memory address. This information is stored in a Demapper RAM which is configured by software as TDM connections are added or torn down.
- For PDU traffic, assemble all 16 slots which make up the PDU into a single 512-byte PDU. Then forward this entire PDU word to the row buffer mapper logic. The PDUs are assembled prior to forwarding them to the row buffer so that the row buffer can write the entire PDU to the row buffer memory in a single clock cycle. This will provide the maximum possible write-side memory bandwidth to the row buffers. This is the most critical constraint of the iTSE implementation, being able to write 12 entire PDUs to a single row buffer in 6 link slot times (12 core clock cycles).
- For Request Elements, assemble the 3-slot block of REs into two 48-bit REs and forward them to the Request Parser module.

The detailed description of this module is provided in Section 4.3.1.2.

3.1.3 Row Buffer Mapper

This module is responsible for mapping traffic which is received from the Data Stream Demappers into the row buffer memories. The feature highlights of this module are:

- FIFO the TDM traffic as it's received from the Data Stream Demappers. Then write it to the row buffer. The row buffer memory address is actually pre-configured in the Demapper RAM within the Data Stream Demapper module. That module will forward the address to the row buffer mapper along with the TDM slot data.
- Write PDU traffic from the Data Stream Demappers to the row buffers. The Row Buffer Mapper will compute the address within the row buffer where each PDU will be written. PDUs will be written into the row buffers starting at address 0 and then every 16-slot address boundary thereafter, up to the maximum configured PDU addresses for the row buffer.

The detailed description of this module is provided in Section 4.3.1.3.

3.1.4 Row Buffer

This module simply contains the row buffer memory elements. The requirements are:

- Provide double buffered row storage which will allow one row buffer to be written during row N while the row data which was written during row N-1 is being read out by the Data Stream Mapper.
- Each row buffer must be capable of storing 1536 slots of data. This will allow the row buffer to store 96 PDUs or 1536 TDM slots or a combination of the two traffic types. Request elements and Link Overhead slots are NOT sent to the row buffer, therefore the row buffer does not need to be sized to accommodate the entire 1700 input link slots.
- The row buffer write port must be $16 \times 36 = 576$ bits wide. It must support writing of only one 36-bit slot (TDM data) or writing of an entire 576-bit word (PDU data) in a single clock cycle.

The detailed description of this module is provided in Section 4.3.1.4.

3.1.5 Request Arbitration

The request arbitration consists of 2 components: (1) a centralized Request Parser module and (2) a Request Arbitration module for each of the output links.

Request Elements are extracted from the input slot stream by the Data Stream Demapper modules and then forwarded to the Request Parser. The Request Parser (which is summarized in Section 3.2.2) will forward the 48-bit request elements to the Request Arbitration modules via two request busses. Each request bus may contain a new request element each core clock cycle. This timing will allow the Request Arbitration logic to process all 13 request sources in less than 8 core clock cycles. The 13 request sources are the 12 input data streams and the internal Multicast & In-Band control messaging module.

The Request Arbitration module will monitor the two request element buses and read in all request elements which are targeted for output link the Request Arbitration module is implementing.

Requirements for this Request Arbitration module are:

- Provide buffering for up to 24 request elements.
- When a new request element is received store it in a free RE buffer. If there are not any free buffers, then replace the lowest priority RE which is already stored in a buffer with the new RE if the new RE is a higher priority. If the new RE is equal to or lower in priority than all REs currently stored in the buffers then discard the new RE.
- On the output side, when the Data Stream Mapper module is ready to receive the next RE, forward the highest priority RE which is stored in the RE buffers to the Data Stream Mapper module. If the RE buffers are empty, then forward an "Idle" RE.

The detailed description of this module is provided in Section 7.

3.1.6 Data Stream Mapper

This module is responsible for inserting data into the outgoing serial data links. The feature highlights of this module are:

- Mapping of the output link slots. This means based on the output slot number determine if the traffic is TDM, PDU, Request Element, or test traffic. The determination is based on the contents of the Mapper RAM.
- For TDM traffic, determine the row buffer memory address. This information is stored in a Mapper RAM which is configured by software as TDM connections are added or torn down.
- For PDU traffic read one slot at a time from the row buffer. The row buffer memory address is stored in the Mapper RAM by software. If the target PDU is not valid (i.e., a PDU was not written to that row buffer location during the previous row time), then transmit the idle pattern, this will insure that a data PDU is not duplicated within the switch.
- For Request Elements, assemble the 3-slot block of REs from two 48-bit REs. The REs are read from the Request Arbitration module.
- For test patterns, insert the appropriate test pattern from the Output Link Bus. These test patterns are created by either the Test Pattern Generator or Test Interface Bus modules.

- Support slot multicasting at the output stage. For example, if we're the Data Stream Mapper for output link 3, we will be able to copy whatever any other output link is sending out on the current slot time. This copying is controlled via the Mapper RAM and will allow the Mapper to copy the output data from another output link on a slot-by-slot basis.

The detailed description of this module is provided in Section 4.3.1.5.

3.1.7 Data Stream Serializer

The feature highlights of this module are:

- Create the output slot stream, data slots are received via the Data Stream Mapper module, overhead slot data is generated internally to this module.
- Split the row data stream into two byte streams for transmission on two Unilink drivers.
- Scramble the output byte stream
- Serialize the output byte stream

The detailed description of this module is provided in Section 11.

3.1.8 Grant Stream Deserializer

The Grant Stream Deserializer works in much the same manner as the Data Stream Deserializer. The primary difference is that the grant data only utilizes a single Unilink receiver, thus eliminating the need for deskewing and deinterleaving to recover a single input serial stream.

Since this serial link will only be one half the data stream rate, there will only be 850 slots per row time.

A single FIFO is used to allow for deskewing of the input serial grant streams for all 12 links.

The detailed description of this module is provided in Section 11.

3.1.9 Grant Stream Demapper

This module is responsible for extracting the data from the incoming serial grant links. The feature highlights of this module are:

- Demapping of the received grant link slots. This means based on the input slot number determine if the traffic is a Grant Element or another kind of traffic. The determination is based on the contents of the Grant Demapper RAM. Note: Traffic other than Grant Elements is TBD.
- For Grant Elements, assemble the 3-slot block of GEs into two 48-bit GEs and forward them to the Grant Parser module.

The detailed description of this module is provided in Section 7.2.3.1.

3.1.10 Grant Arbitration

The grant arbitration operates in an identical manner to the Request Arbitration logic. In fact, this module is identical to the Request Arbitration module, the only difference is that it's processing grant elements in the reverse path instead of request elements in the forward path.

3.1.11 Grant Stream Mapper

This module is responsible for inserting data into the outgoing serial grant links. The feature highlights of this module are:

- Mapping of the output grant slots. This means based on the output slot number determine if the traffic is a Grant Element or test traffic. The determination is based on the contents of the Grant Mapper RAM.
- For Grant Elements, assemble the 3-slot block of GEs from two 48-bit GEs. The GEs are read from the Grant Arbitration module.
- For test patterns, insert the appropriate test pattern from the Output Link Bus. These test patterns are created by either the Test Pattern Generator or Test Interface Bus modules.

The detailed description of this module is provided in Section 7.2.3.2.

3.1.12 Grant Stream Serializer

The Grant Stream Serializer works in much the same manner as the Data Stream Serializer. The primary difference is that the grant data only utilizes a single Unilink transmitter, thus eliminating the need for interleaving the transmit serial stream across multiple output serial streams.

Since this serial link will only be one half the data stream rate, there will only be 850 slots per row time.

The detailed description of this module is provided in Section 11.

3.2 Datapath & Link BW Arbitration (per chip modules)

This group of modules is instantiated only once in the ITSE. These modules provide support functions as part of the implementations of the core data path switching functions.

3.2.1 Link Synchronization & Timing Control

This module provides the global synchronization and timing signals used in the ITSE. Some of its features are:

- Generate transmission control signals so that all serial outputs start sending row data synchronized to the RSYNC (row synchronization) input reference.
- Control the deskewing FIFOs in the Data Stream Deserializers so that all 12 input links will drive the data for slot N at the same time onto the input link bus. Note: this same deskewing mechanism is implemented on the Grant Stream Deserializers.

The detailed description of this module is provided in Section 10.

3.2.2 Request Parser

This module will receive inputs from all 13 request element sources and forward the REs to the Request Arbitration modules via two request element buses. Basically, this module is mapping the 13 parallel RE inputs onto two TDM buses.

The detailed description of this module is provided in Section 7.2.1.1.

3.2.3 Grant Parser

The grant arbitration operates in an identical manner to the request arbitration logic. In fact, this module is identical to the Request Parser module, the only difference is that it's processing grant elements in the reverse path instead of request elements in the forward path.

3.2.4 Link RISC Processor

This module will be a Tensilica processor core (one of two on the ITSE) which will implement these functions:

- Control the ranging synchronization on the input links with the source ITPP. This function only needs to be done on an ITSE which resides within the first stage of the switch fabric.
- Likewise, control the ranging synchronization on the output link grant stream input with the source ITPP (i.e. ITPP generating the grant stream). This function only needs to be done on an ITSE which resides within the last stage of the switch fabric.
- Multicast controller. This Link RISC Processor will handle the Req/Grant processing needed to transmit multicast messages.
- In-band data communications controller. This module will only control the reception and transmission of the in-band communications PDUs. All PDUs will be forwarded to the Configuration RISC Processor which will interpret the messages. This Link RISC Processor will only handle the Req/Grant processing needed to transmit messages.

The detailed description of this module is provided in Section 6.

3.3 Support Modules

This group of modules is instantiated only once in the ITSE. Since the primary function of the ITSE is switching of traffic between the input and output links, we can view modules which are not actively transporting traffic as "support" modules for the switching functions.

3.3.1 Configuration RISC Processor

This is a Tensilica based RISC processor core, it is one of two Tensilica RISC processors which is present in the iTSE. The primary function of this core is to process configuration and status messages from an external (to the iTSE) controller module.

The detailed description of this module is provided in .

3.3.2 System Control

This module will handle all the reset inputs and reset the appropriate internal modules.

The detailed description of this module is provided in Section 12.

3.3.3 Test Pattern Generator & Analyzer

This module will be used for the generation of various test patterns which can be sent out on any slot on the Data Stream or Grant Stream outputs. It will also be capable of monitoring input slots from either the received Data Stream or Grant Stream.

The detailed description of this module is provided in Section 17.1.

3.3.4 Test Interface Bus Multiplexer

This module will allow for sourcing transmit data from the external I/O pins. Also, received data can be forward to the I/O pins. This will be used for testing the iTSE when an iTPP may not yet be available.

The detailed description of this module is provided in Section 17.2.

3.3.5 PLLs

The Unilink PLL is used to create the IF clock needed by the Unilink macros. Within each Unilink macro another PLL will multiply the IF clock up to the serial clock rate.

The Core PLL is used to create the clock used by the iTSE core logic. This core clock is expected to be around 250 MHz.

The detailed description of these PLLs is provided in Section 9.

3.3.6 JTAG

The JTAG interface is used for two purposes: (1) boundary scan testing of the iTSE at the ASIC fab and (2) Debug interface for the Configuration RISC Processor. Note: the Link RISC Processor will not have a debug interface, it will be implementing finite state machines so we want to keep it as small as possible.

The detailed description of this module is provided in Section 17.4.

Proprietary and Confidential Information of Onex Communications Corporation

3.4 Internal Datapath Buses

RESERVED

[illegible]

4 Data Path Description

This section will describe how the switch data path is implemented within an iTSE. There are two types of data which may be switched through the iTSE, TDM data and Data PDUs (which can carry an ATM cell or a fragment of an IP packet). This section will focus on TDM and unicast Data PDU switching. Multicast Data PDU switching is described in Section 5.

The switching mechanism will operate on a row by row basis. This means TDM or Data traffic on an input link during any given row time may be switched to any output link and/or slot time within the same given row time. The structure of the iTSE is a Time-Space-Time division fabric. This concept is illustrated in Figure 4-1.

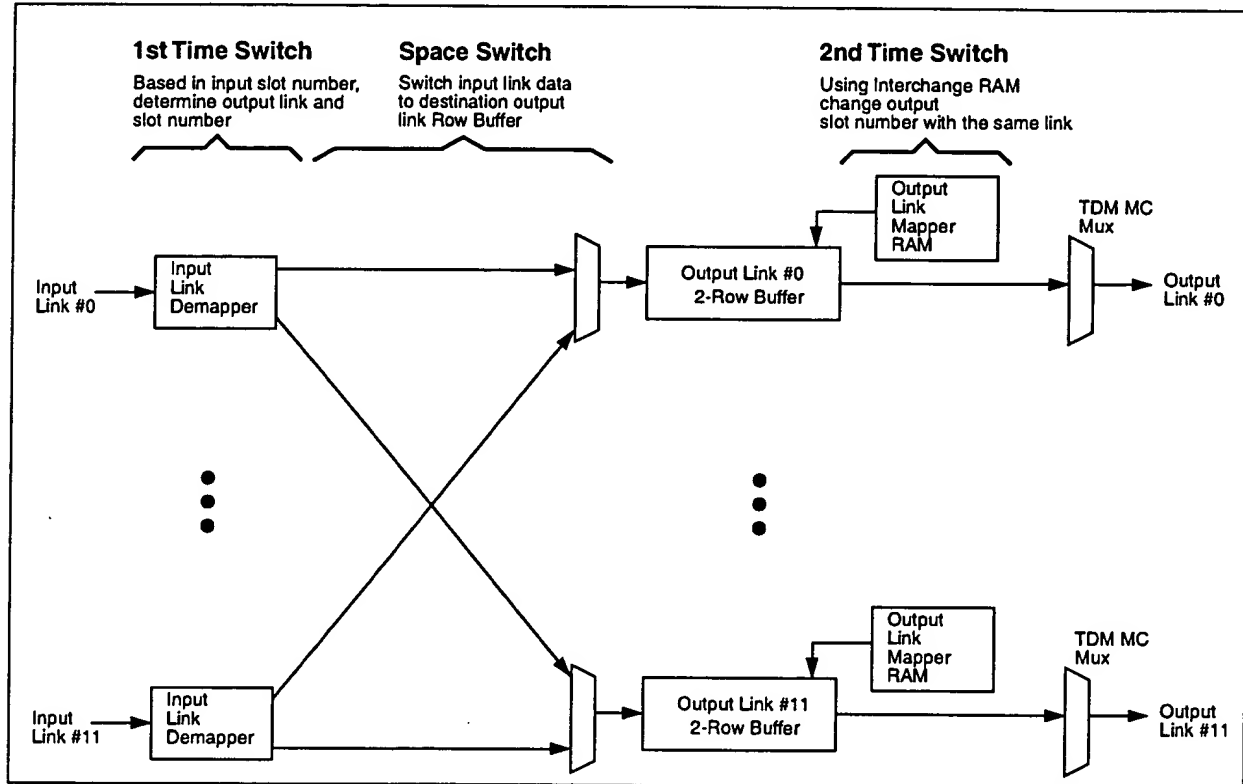


Figure 4-1: iTSE Time-Space-Time Switching Fabric

The T-S-T switching concept only applies to TDM traffic. For data PDUs, the implementation of the iTSE allows the data PDUs to reside in any group number within the row. This means that for data traffic the iTSE switch fabric can simply be viewed as a Space switch.

Since the iTSE switches traffic on a row by row basis, a 2-Row Buffer store is required. While one row is being received and written into one of the 2 row buffers, the other row buffer (which contains data received during the previous row time) is being played out to the output link.

TDM traffic is switched based on how the Input Link Demapper RAM, of which there is one per input link, and the Output Link Mapper RAM, of which there is also one per output link, are configured. Reference Figure 4-16 and Figure 4-19 for where these RAMs are implemented. These RAM are configured via the internal RISC processor, which in turn gets the configuration messages from an external software module which is determining the switching path for each new TDM stream which is added to the switch fabric.

Data traffic is switched as 16-slot "PDUs" through the switching fabric. Each PDU will include a self-route tag which identifies the path it will take through the switching fabric. Data PDUs enter the switching fabric based on scheduling algorithms which are running on each input Port Processor chip. Since these scheduling algorithms are independent, there is not any synchronization between the iTPPs. This could result in the iTPPs sending more data PDUs to a single output row buffer than it can store, which would result in the switch dropping the excess PDUs. In order to prevent this situation from occurring, the concept of "arbitration" for the PDU data path has been introduced to the iTSE architecture. With this arbitration scheme, the source iTPPs will send request messages to the destination iTPPs. The destination iTPPs will in turn reply with a grant message, via a separate out-of-band control channel which is implemented

Proprietary and Confidential Information of Onex Communications Corporation

as a full reverse overlay network within the switch fabric, to the source iTPPs. If the source iTPP receives a grant for a request, it will be able to send the data PDU in the next row and be certain that the PDU will get to the destination iTPP and not be lost within the switch fabric due to buffer overflow.

This arbitration scheme is described in detail in Section 7. This arbitration scheme, in concert with the implementation of the data path, will guarantee that no data is lost within the switch fabric.

4.1 Constraints on Switch Configuration

This section summarizes the constraints on iTSE usage which, if met, will guarantee that no data is lost within the switch fabric.

TDM Switching Constraints -

- TDM traffic may not be transported on the last 36 slots of the row.
- In any given slot, no more than 2 TDM slots on any of the 12 input links may be destined for the same row buffer input TDM FIFO (see Figure 4-16 for where the TDM FIFOs reside).
- In any given 16-slot period, no more than 18 total TDM slots may be destined for the same row buffer input TDM FIFO.

4.2 iTSE Data Path Timing

Figure 4.2 illustrates the basic timing for writing incoming data into the Row Buffers. Writing of incoming data to the Row Buffers is the critical timing path because it's possible that as many as 14 traffic sources will simultaneously send traffic destined to the same output Row Buffer. The 14 traffic sources are the 12 Input Link Demappers, the Multicast Controller, and the Control Message Controller.

The timing concept will operate on a "group" basis, where a group will define the 16-slots which may contain a data PDU. The input links associated with a specific switch stage must all have the same configuration for data PDU slots. This means that the data PDU slots must all be defined to line up on the same slot boundaries. This definition is done by way of the Input Link Demapper RAM configuration.

For example, let's say input links 0, 1, and 2 are all configured to be within the same switch stage. In addition, let's say that link 0 contains 3 PDUs, link 1 contains 1 PDU, and link 2 contains 2 PDUs. One possible configuration for where the PDUs are placed on each of these 3 links would be:

- Link 0 - 1st PDU in slots 2 through 17, 2nd PDU in slots 20 through 35, 3rd PDU in slots 36 through 51.
- Link 1 - 1st PDU in slots 20 through 35.
- Link 2 - 1st PDU in slots 2 through 17, 2nd PDU in slots 36 through 51.

Observed that the 16-slot PDUs for all input links must always fall within the same link slot boundaries. Links which carry fewer PDUs may use the unused PDU slot areas for carrying TDM traffic.

The 16-slot group time period is then divided into two half-group periods, which are called "1st 1/2 group period" and "2nd 1/2 group period". During group number N, if an incoming link is carrying a data PDU, the 1st half of the data PDU will be assembled into one large 256-bit word (8 slots * 32 bits/slot). Half way into group number N, the Row Buffer Mappers will be given a start signal which tells them that they may now write the first half of the PDUs into the upper part of the row buffers. This row-buffer write period may take up to 12 cclk cycles. While the 1st half of the group is being written to the row buffers, the 2nd half of the group is being assembled into another larger 256-bit word (8 slots). At the end of group number N, the 2nd half of the PDU will be written into the lower part of the row buffers. This timing is illustrated in Figure 4-2.

If an incoming link is carrying TDM data, the TDM slot data will be written directly into the appropriate TDM FIFO. The contents of the TDM FIFOs are then written to the row buffers during periods where PDU data is not being written.

The internal core clock for this datapath logic will be running at 2x the slot rate, i.e., there will be 2 clock cycles available to process each received slot of data. Thus, on average, 7 slot times (14 cclk cycles) are used to write up to 14 PDUs of data to the Row Buffer RAM. The remaining 9 slot times (18 clock cycles) will be used to unload the TDM FIFOs and write their contents to the Row Buffer RAM.

This timing is illustrated in the figure below. Because there are only 2 core clock cycles available per incoming slot time, only 2 TDM slots of data may be written to any single TDM FIFO in that single slot time. This will be a

Proprietary and Confidential Information of Onex Communications Corporation

4.3 Datapath Core Implementation

A block diagram of the data path core of the iTSE is shown below.

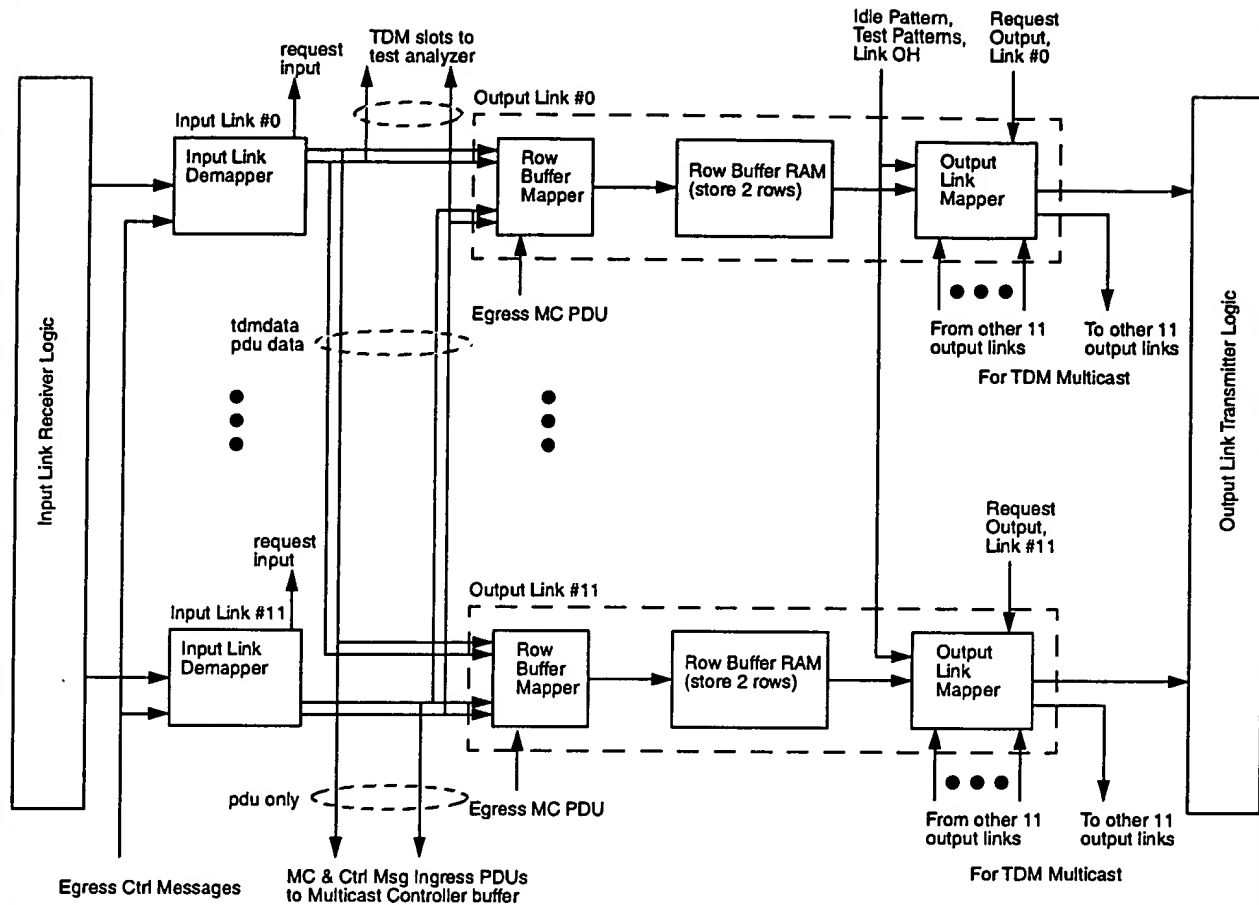


Figure 4-3: Switch Datapath Block Diagram

This Datapath core will be implemented with the hierarchy shown in Figure 4-4.

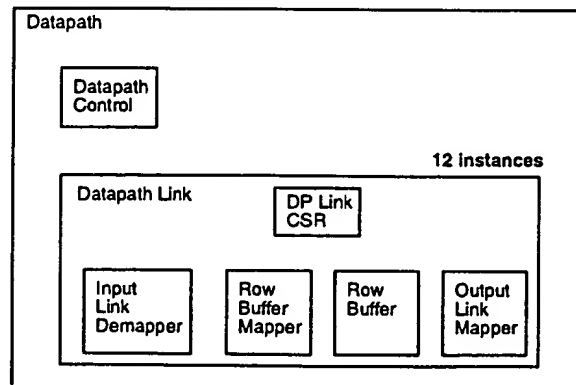


Figure 4-4: Datapath Module Hierarchy

Proprietary and Confidential Information of Onex Communications Corporation

4.3.1 Datapath Link Module

A Datapath Link module is comprised of the modules which will make up a single switch datapath link. The iTSE Datapath module will instantiate 12 of these Datapath Link modules.

As shown in the Figure 4-5, a Datapath Link module will instantiate these modules:

- Input Link Demapper
- Row Buffer Mapper
- Row Buffer RAM
- Output Link Mapper
- Datapath Link CSR

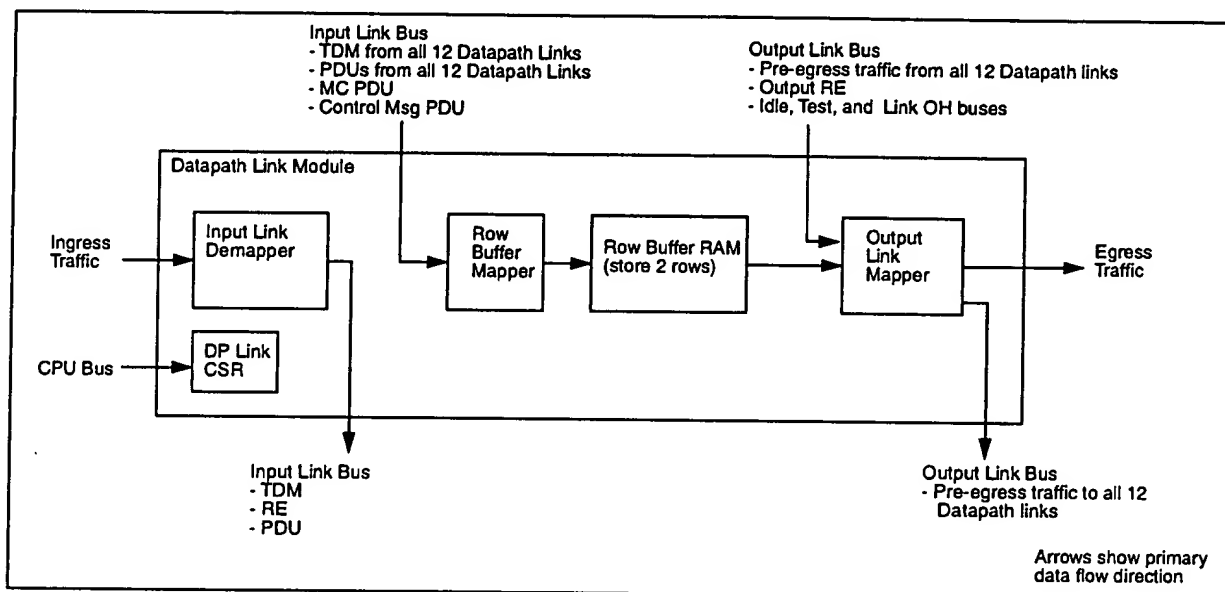


Figure 4-5: Datapath Link Module Interfaces

4.3.1.1 Interface I/O Signals & Timing

Because there are a large number of I/O ports associated with a Datapath Link module we'll summarize them in the table below so that the reader has a feel for the I/O signals prior to describing the implementation of each of the modules which make up a Datapath Link. Figure 4-5 illustrates the interfaces of a Datapath Link module.

Ingress Traffic Timing -

Figure 4-6 illustrates the timing for incoming data slots. The iTSE implementation will require cclk to be at least twice the incoming slot rate. This will insure that the iTSE will have a minimum of two cclk cycles to process each incoming slot of data. The two cclk cycles per slot are split into two phases in order to identify which cclk edge the incoming islot_num and islot_data signals are changing.

The iTSE will allow cclk to be asynchronous to the serial link clocks, the only requirement is that cclk be chosen such that there are always a minimum of two cclk cycles per slot. Because cclk may be asynchronous to the incoming link, there may more that 2 cclk cycles for any given input slot. In this case both islot_phase0 and islot_phase1 will both be deasserted for all but the first two clock cycles for each input data slot. This case is shown during islot_num = 2 in Figure 4-6. Whenever an extra timing adjust clock cycle is inserted, the signal islot_phase2 will be asserted.

Notes:

- The islot_row_end would normally occur during the last slot of the link. But we do have the option of speeding up the link (for characterization in the lab), which would result in the link having more than 1700 slots per row. In this

Proprietary and Confidential Information of Onex Communications Corporation

scenario, islot_row_end will be asserted starting at slot 1699 and remain asserted until slot 0 of the next row.

- If link synchronization is lost, islot_phase0 and islot_phase1 will remain deasserted and islot_phasez will be asserted.

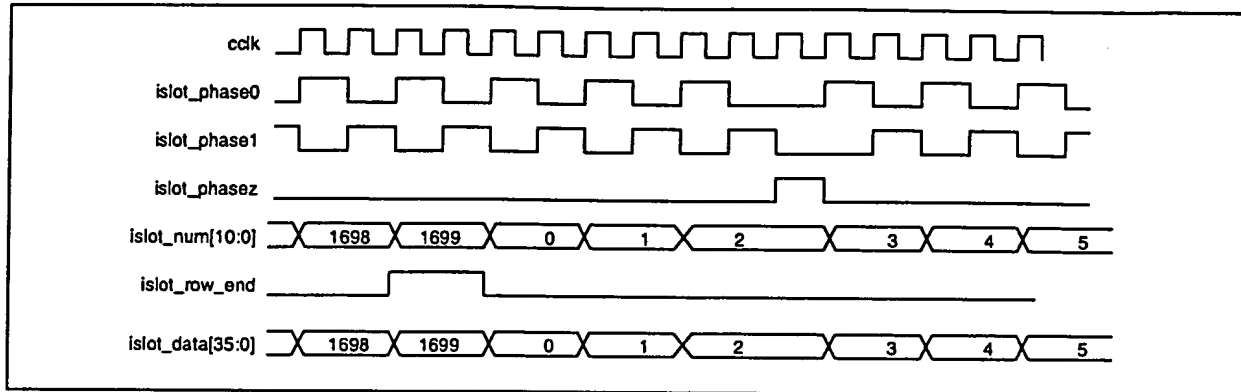


Figure 4-6: Ingress Traffic Timing

Input Bus Timing -

For the input bus we'll show these timing diagrams:

- Driving received TDM slots onto the Input Bus.
- Driving received PDUs onto the Input Bus.
- Driving received REs onto the Input Bus.

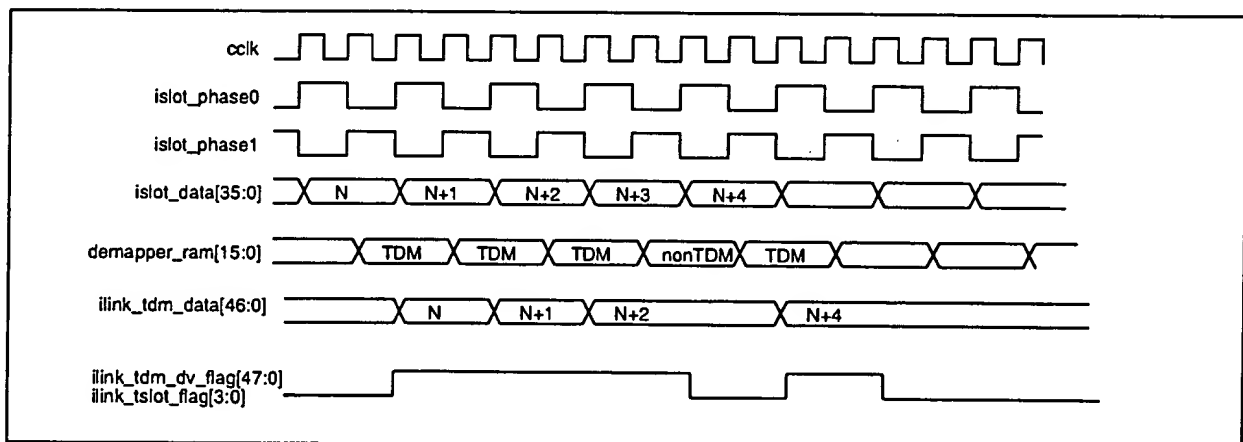


Figure 4-7: Input Link Bus, TDM Timing

Proprietary and Confidential Information of Onex Communications Corporation

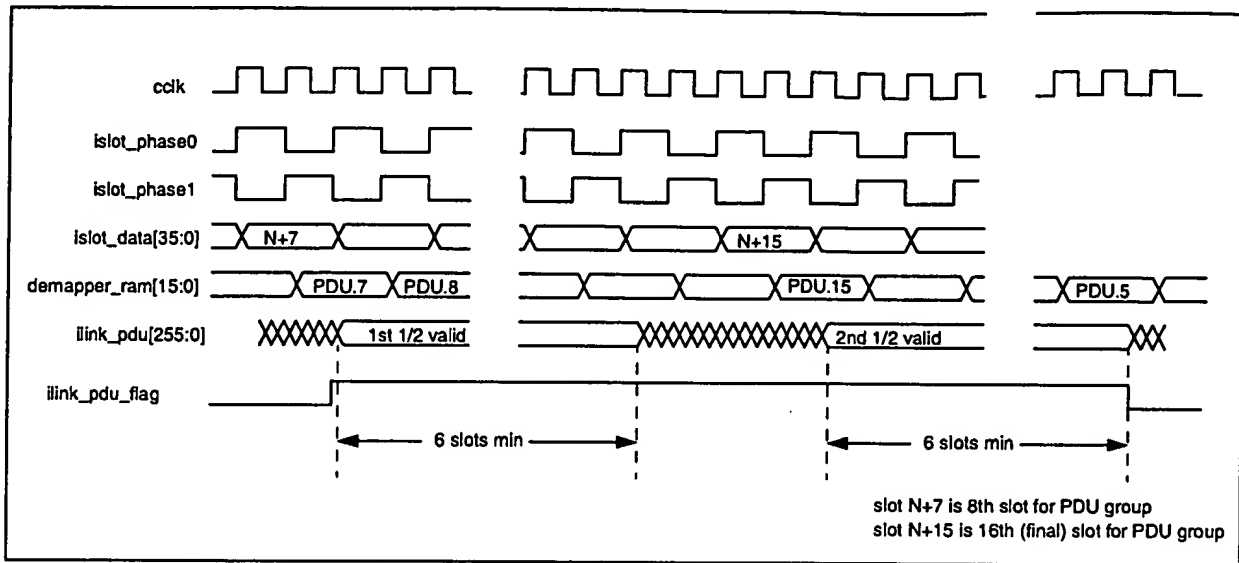


Figure 4-8: Input Link Bus, PDU Timing

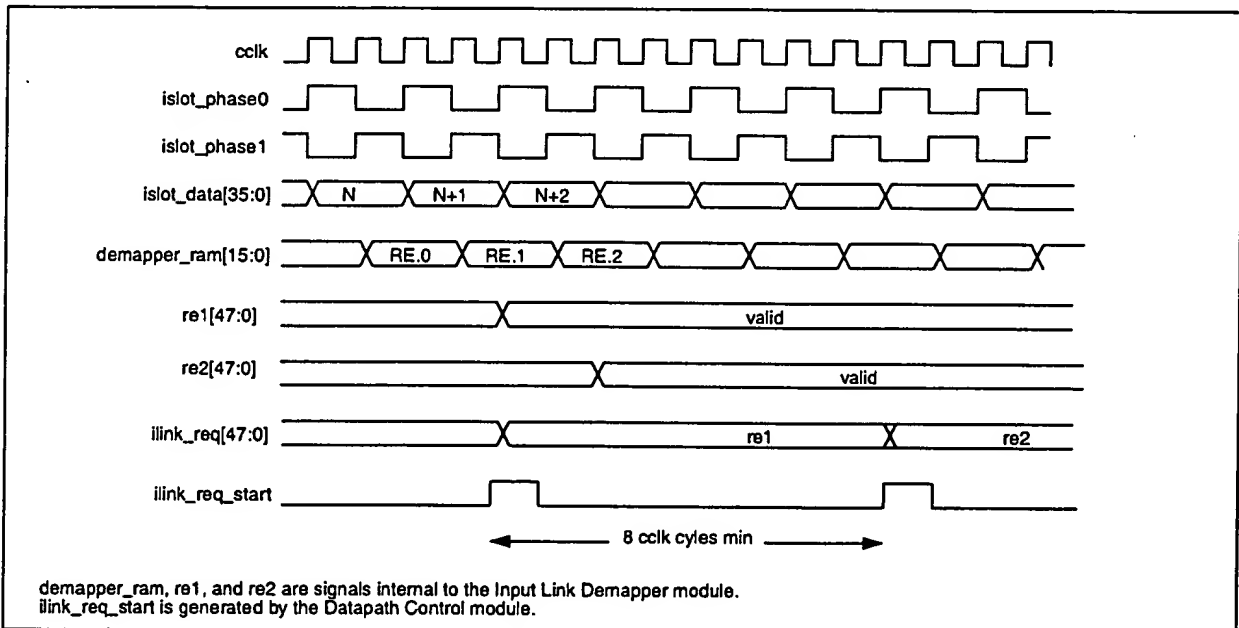


Figure 4-9: Input Link Bus, Request Element Timing

Proprietary and Confidential Information of Onex Communications Corporation

4.3.1.2 Input Link Demapper

This module is instantiated once per Link module. A high-level block diagram is shown in Figure 4-10.

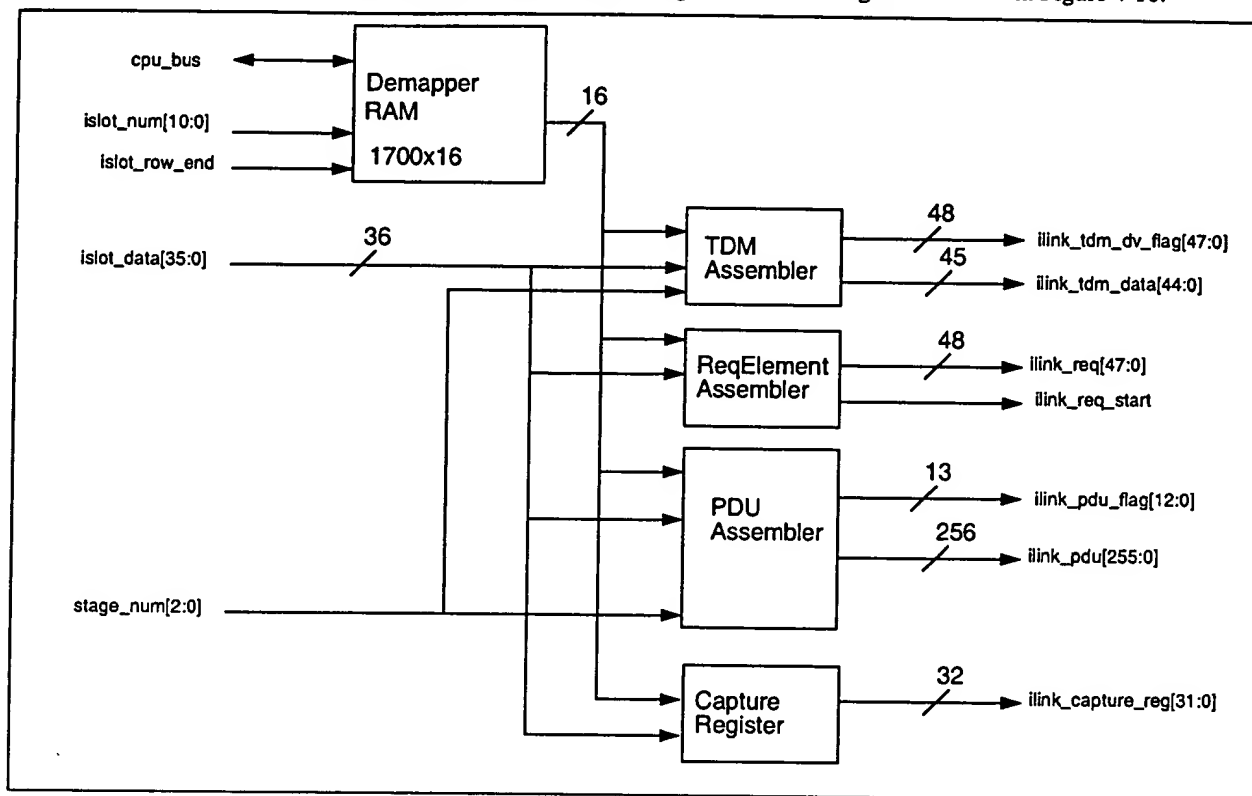


Figure 4-10: Input Link Demapper Block

The inputs to this module are the CPU Bus and the Ingress Traffic interface. These I/O interfaces are described in Section 4.3.1.1.

4.3.1.2.1 Demapper RAM

The Demapper RAM determines how each slot on the input link is being used. The RAM will be configured by the CPU to do two primary functions:

- Define the structure of the input link. This means identifying whether the input slot is carrying TDM traffic, a portion of a data PDU, a portion of a Request Element, a Test Traffic slot, or is slot which should be ignored.
- For TDM traffic, it also specifies the destination row buffer and the address within that row buffer. This is the Time-Space cross-connect mapping function.

The islot_num input is incremented once for each incoming slot and will be used as the address for this RAM. This input slot number will start at 0 for the first slot and then simply be incremented once for each new input slot up to the maximum number of slots in the row.

Since there will always be at least 2 cclk cycles per input slot, the accessing of the RAM is split into two phases. During phase 0 the RAM will be addressed using the islot_num input, the output of the RAM will be registered at the end of phase 0 so that it will remain valid for the following 2 cclk cycles. During phase 1, the RAM may be written to or read from by the CPU. This timing is illustrated in Figure 4-11. The RAM will be implemented with a 1 write, 1 read port memory cell.

Proprietary and Confidential Information of Onex Communications Corporation

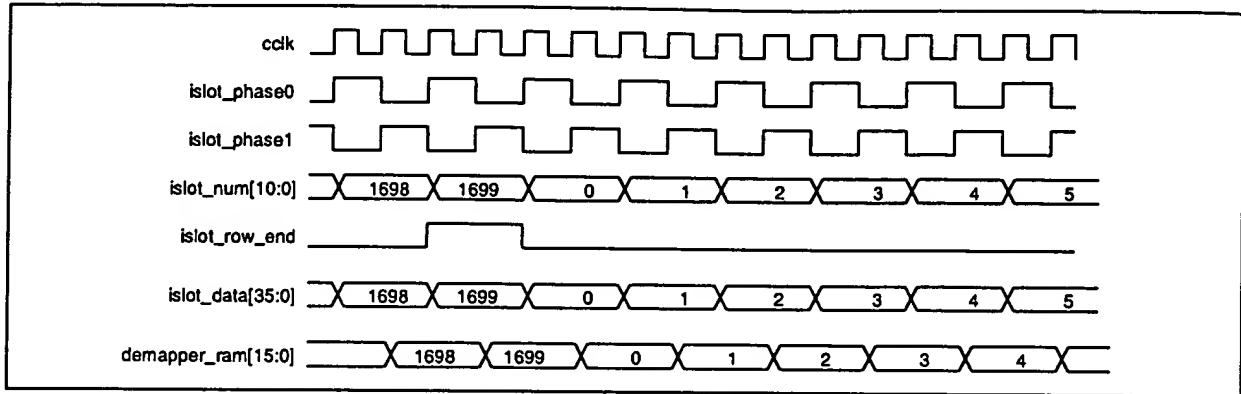


Figure 4-11: Demapper RAM Timing

The Demapper RAM is 16-bits wide. The first bit identifies whether or not the slot contains TDM traffic. If so the remaining 15-bits identify the destination row buffer and address within that row buffer. If the slot is not TDM traffic, then the remaining 15-bits are used to identify what the slot could be used for. This structure of the RAM is shown in the figures below.

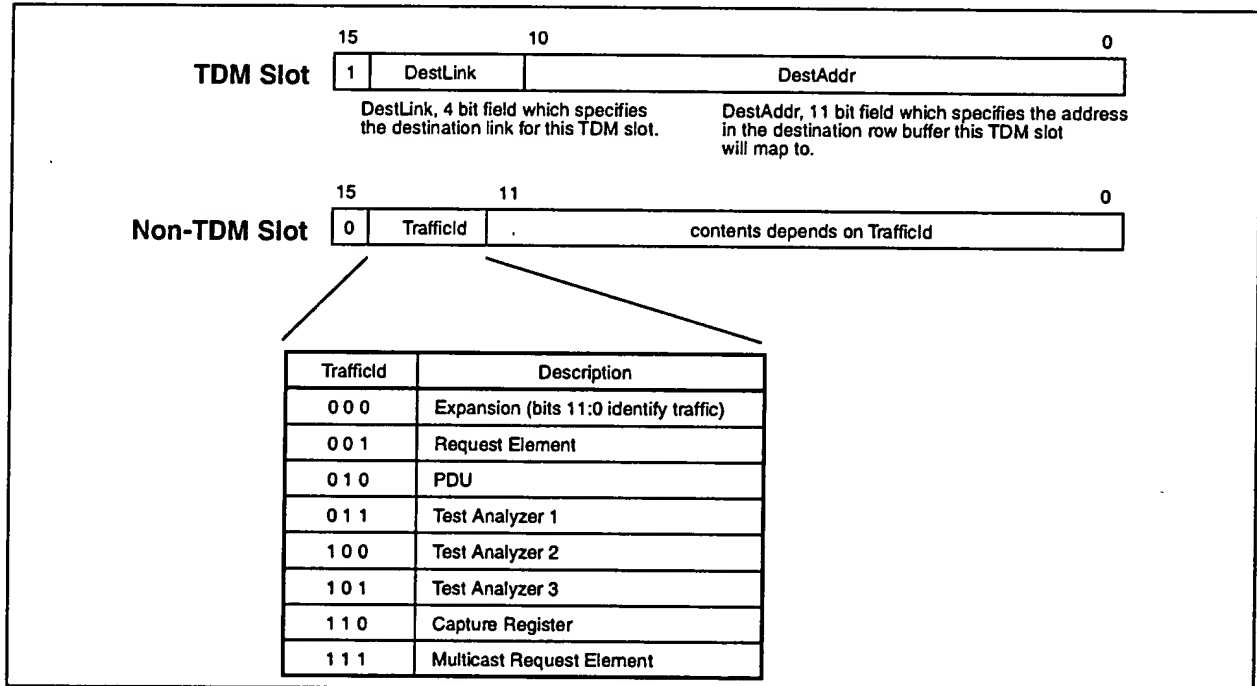


Figure 4-12: Demapper RAM Structure, TDM vs Non-TDM Slot

Proprietary and Confidential Information of Onex Communications Corporation

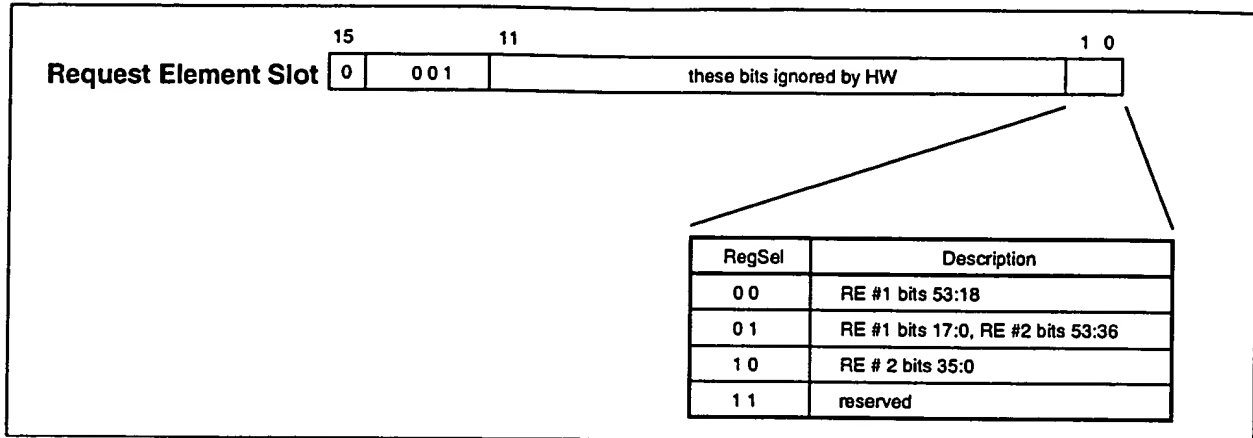


Figure 4-13: Demapper RAM Structure, Request Element Slot

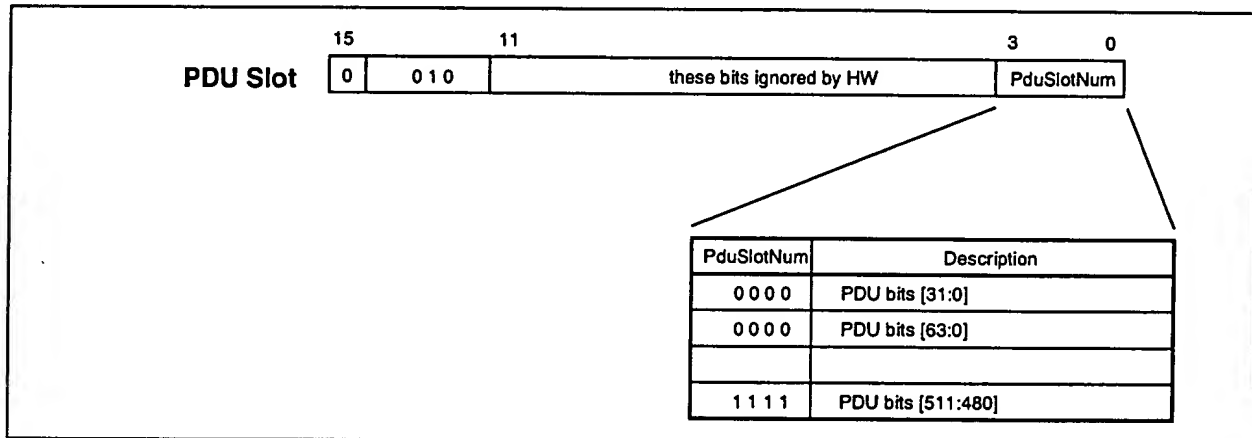


Figure 4-14: Demapper RAM Structure, PDU Slot

Table 4-1: Expansion Slot Definitions

Demapper Ram [11:0]	Traffic Type	Comments
0	idle	Unused ingress slot, will be ignored by Demapper logic.
1	BIP-36	Bit Interleaved Parity for all slots from slot 0 or previous BIP-36 slot.
2	LOH Status	Link Overhead slot which contains status information from the remote connection in bits 28:24.
2-4095		reserved for future use

4.3.1.2.2 TDM Assembler

The TDM Assembler module simply latches both the TDM slot data (islot_data) and the output of the Demapper RAM when the incoming slot is a TDM slot. It then generates signals (ilink_tdm_dv_flag[47:0]) which inform the Row Buffering logic that TDM data is available.

The timing diagram below illustrates the receiving of 4 TDM slots on the input link. Slot numbers N, N+1, N+2, and N+4 are the TDM slots.

Proprietary and Confidential Information of Onex Communications Corporation

Also, non-TDM slots which don't fall into the Request Element or PDU category will also be driven out on this TDM bus. The TrafficId bits are decoded be driven out on the ilink_slot_flag[3:1] ports.

4.3.1.2.3 Request Element Assembler

Because the request element (RE) size is 54 bits, it cannot fit into a single slot. Therefore, in order to minimize the number of slots required to carry up to 96 REs per row we'll define a structure which will pack 2 REs into 3 slots. The request element assembly block will contain 2 54-bit buffers which will hold the two REs from the 3 slot structure.

As shown in Figure 4-13, the 2 Lsb's of the Demapper RAM will specify which buffer register the current 36-bit slot will be written to. In this timing example of Figure 4-9, the 3 slots which contain the request payloads are slot numbers N, N+1, and N+2. The ilink_req_start output will inform the request arbitration logic that it can now start processing the new request element which is available on the ilink_req bus.

4.3.1.2.4 PDU Assembler

Data PDU slots will be forwarded to the PDU assembler where they will be assembled into a 256-bit wide words (8 slots * 32 bits/slot). This 8 slots will be the two halves of a 16-slot PDU. After each half-PDU is assembled, the 256-bit half-PDU is available to the row buffer memories so that the entire half-PDU can be written into the row buffer RAM in an single write cycle.

The PDU assembler must be able to store up to 14 slots of data, 8 slots for the half-PDU, plus another 6 slots to store the first 6 slots of the next incoming half-PDU while the main half-PDU store buffer is waiting for the row buffer RAM to write the half-PDU to the memory.

The appropriate ilink_pdu_flag is asserted for the output link this PDU should be forwarded to. For each PDU received on the input link, the flag to set is determined by the appropriate 4-bit field from the self-route tag in the PDU header. The stage_num control input determines which 4-bits to use from the self-route tag. This ilink_pdu_flag field is one bit per output link (as opposed to using a 4 bit encoded field) so that the Row Buffer Mapper doesn't need to decode a 4-bit field.

The mapping of input PDU slots to the 512-bit complete PDU is shown below:

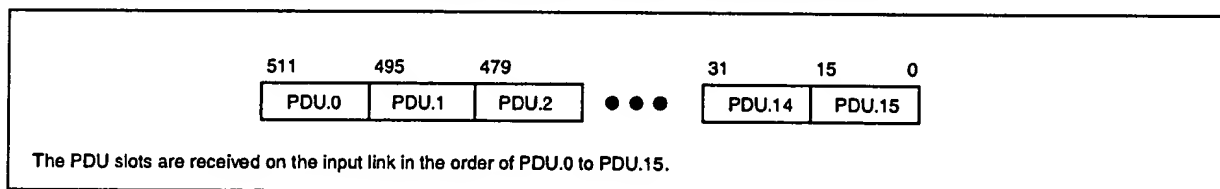


Figure 4-15: PDU Format

Parity Chec k -

RESERVED

4.3.1.2.5 Capture Register

There will be a single 32-bit register which can be used to capture any slot of data from the incoming link. The CPU will be able to read this register at any time. Whenever the capture register is written, the new value will be compared against the previous value and if any bit is different, a LinkCaptureReg IRQ will be generated.

It is expected that this will normally be used to read the Link Overhead Mailbox slot.

In addition to capturing the 32-bit LOH mailbox slot, this module will also have a 4-bit register to capture the LOH status slot. The only bits capture off this status slot are bits 28:24. As is done with the capture register, if any of the 4 bits is different than the previous values, a LinkSyncMsg IRQ will be generated. It is expected that this LOH status slot will contain status information from the remote device connected to this incoming link.

Proprietary and Confidential Information of Onex Communications Corporation

4.3.1.3 Row Buffer Mapper

This module controls the writing of data from the input links to the row buffer. There are two types of data which are written to the row buffer, TDM slots and DATA PDUs. The TDM slot data must be written to the row buffer on a slot by slot basis, i.e., no dependencies between TDM slots is assumed. The DATA PDU is 16-slots wide and in order to keep up with the incoming data rate, must be capable of being written to the row buffer in a single clock cycle.

A high-level block diagram is shown in Figure 4-16.

Design Considerations -

RESERVED

Proprietary and Confidential Information of Onex Communications Corporation

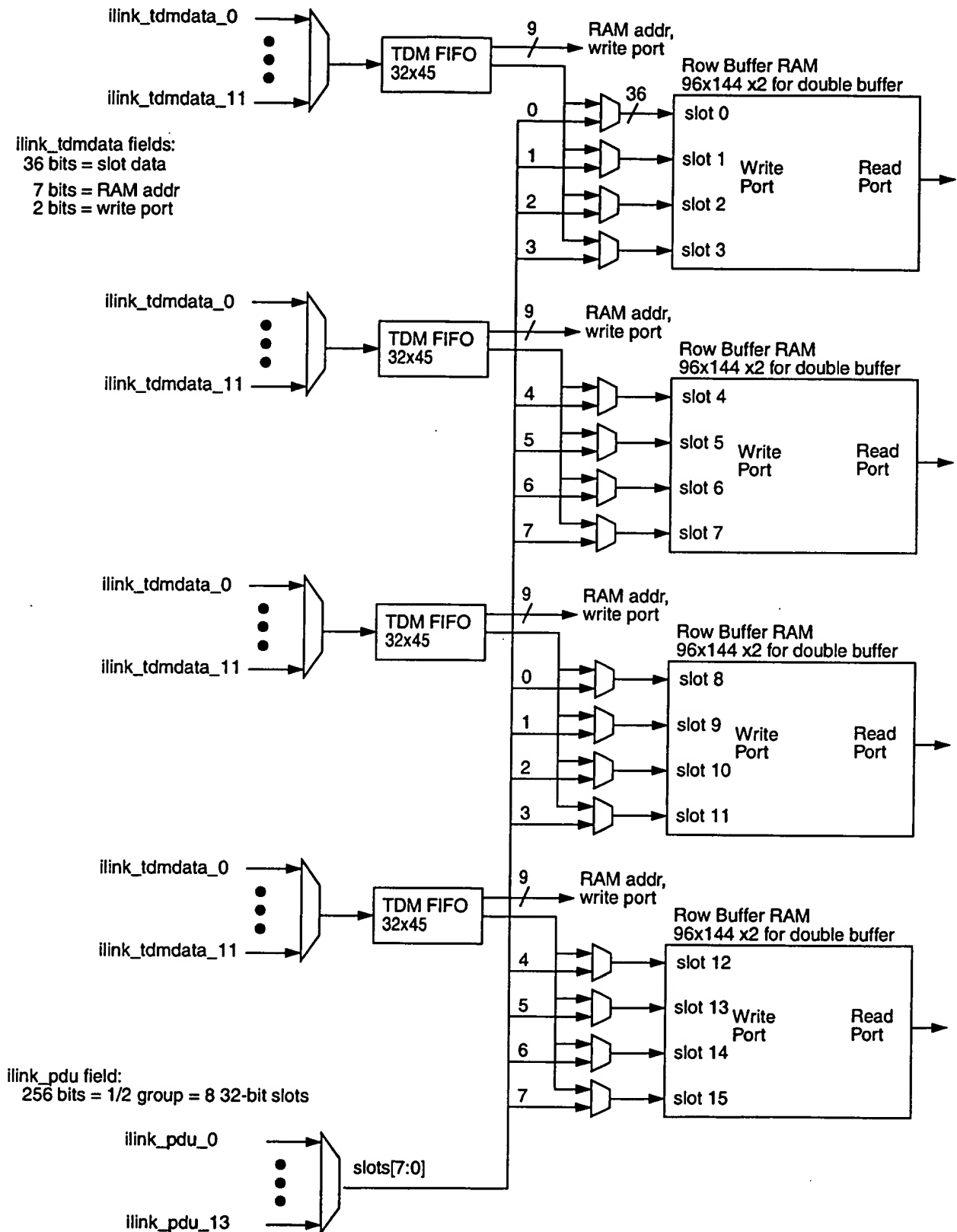


Figure 4-16: Row Buffer Mapper (data path only)

4.3.1.4 Row Buffer

The double-buffered row buffers are used to store the traffic for the output link on a row-by-row basis. This means traffic received on one of the 12 input links during row N will be stored in a row buffer and will then be transmitted on the output link during row N+1.

The row buffers are "double-buffered" in order to support the simultaneous reception and storage of traffic being received during the current row period and the transmission of traffic which was received during the previous row period. At the row boundary (as indicated by the row_toggle input signal) the row buffers will swap.

The row buffers are sized to be 96 PDUs deep. Since each PDU is 16 slots wide, the total storage capacity for a single row buffer is 1536 slots. The observant reader will realize this slot capacity is not large enough to hold an entire row of data from the input link. The input link has a capacity of 1700 slots, but we're sizing the row buffer to only hold 1536 slots. The reasoning for this is:

- 20 slots of the input link are the Link Overhead which doesn't need to be stored in the row buffer.
- If the link is configured to carry 96 PDUs of data traffic, then 144 input link slots will be required to carry the request elements. Request elements are not stored in the row buffers, they are forwarded directly to the arbitration logic by the Input Link Demapper module.

The block diagram of the row buffer is shown in Figure 4-18. The row buffer must be capable of writing an entire PDU (16 slots) of data in a single clock cycle, therefore multiple memory elements must be used in order to get this data bus width.

The address organization of the 96x576 row buffer is shown below:

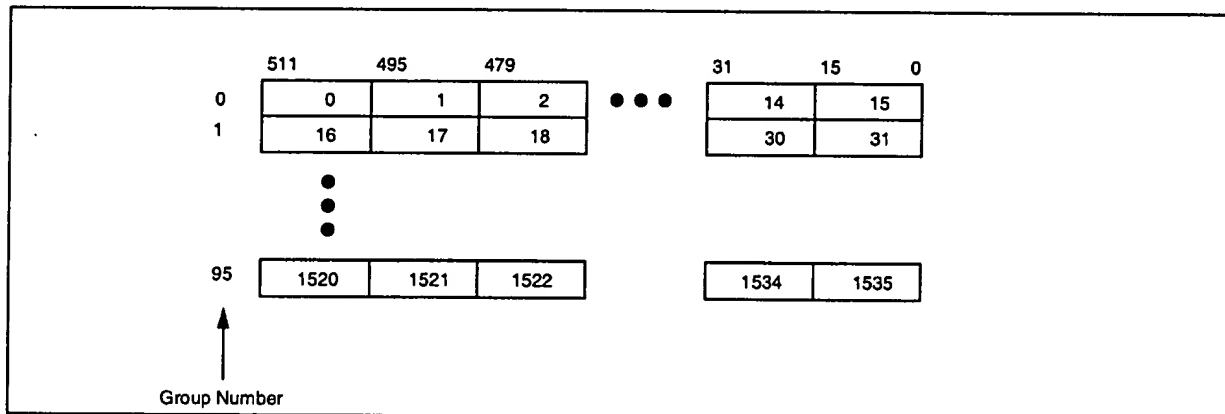


Figure 4-17: Row Buffer Memory Addressing

PDUs will always be stored on an address boundary which is a multiple of 16.

4.3.1.4.1 PDU Fill Status Logic

The function of this module is to monitor the reads from the row buffers by the Output Link Mapper module and indicate to whether or not the an address location being read contains valid PDU data. A location contains valid PDU data if it was written to during the previous row time.

When the Output Mapper logic reads a PDU location which does not contain a valid PDU, it will transmit the idle pattern in place of the PDU slots.

Proprietary and Confidential Information of Onex Communications Corporation

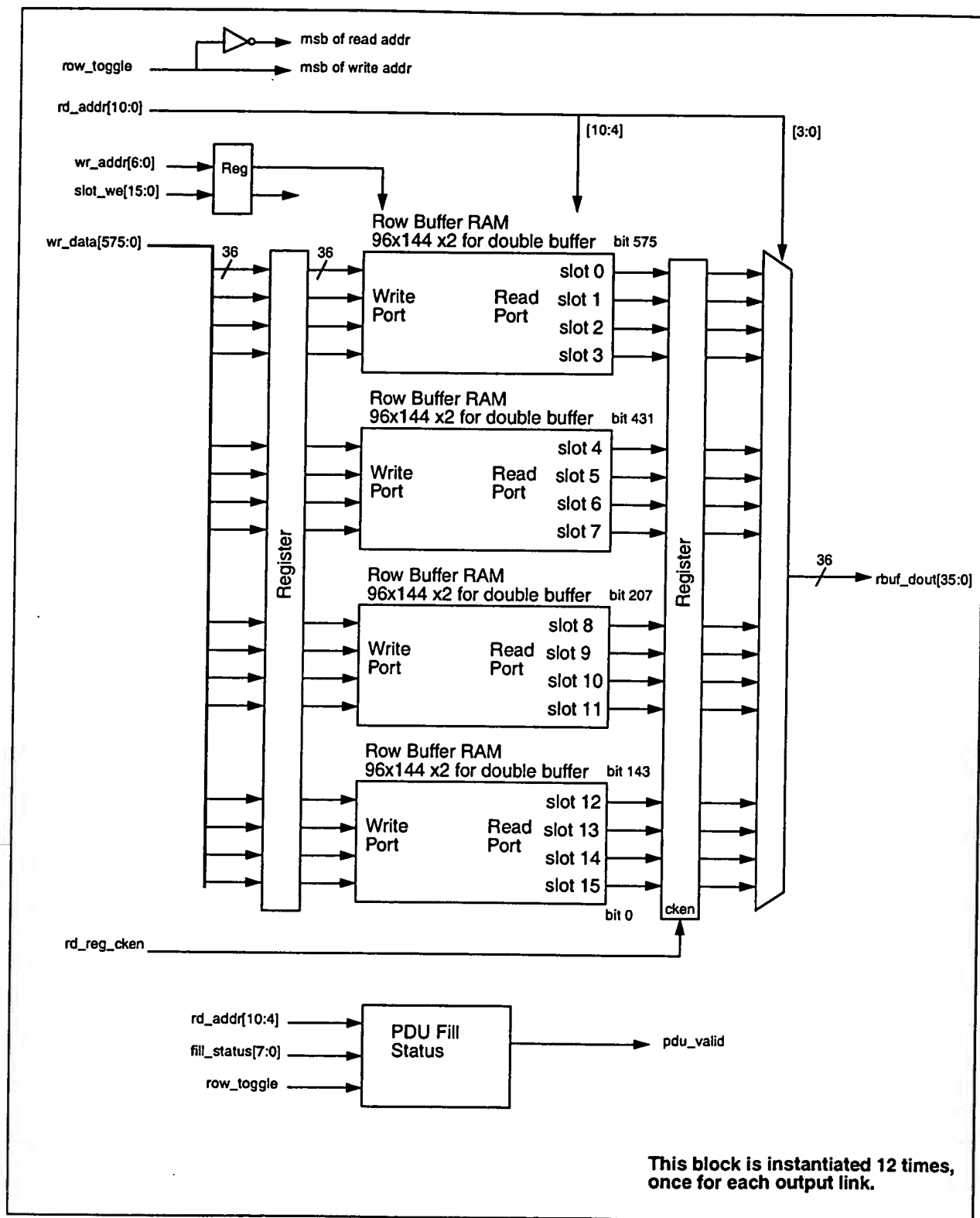


Figure 4-18: Row Buffer Block Diagram

Proprietary and Confidential Information of Onex Communications Corporation

4.3.1.5 Output Link Mapper

The Output Link Mapper module is responsible for mapping the transmit payloads onto the outgoing link slots. The sources of internal data which may be mapped onto an output link slot are:

- TDM or PDU data from the row buffer memory.
- Request Elements from the arbitration module.
- A copy the outgoing slot data from one of the other 11 links. This is the way multicasting of TDM data will be accomplished.
- Idle patterns for unused slots.
- Test traffic from the I/O pins or internal test traffic generators.

This module is instantiated 12 times, once for each output link. A block diagram is shown in the figure below.

The module basically consists of two main functional blocks: (1) a Mapper RAM which identifies what data should be placed into each output link slot and (2) logic which multiplexes the various internal data sources onto the output link.

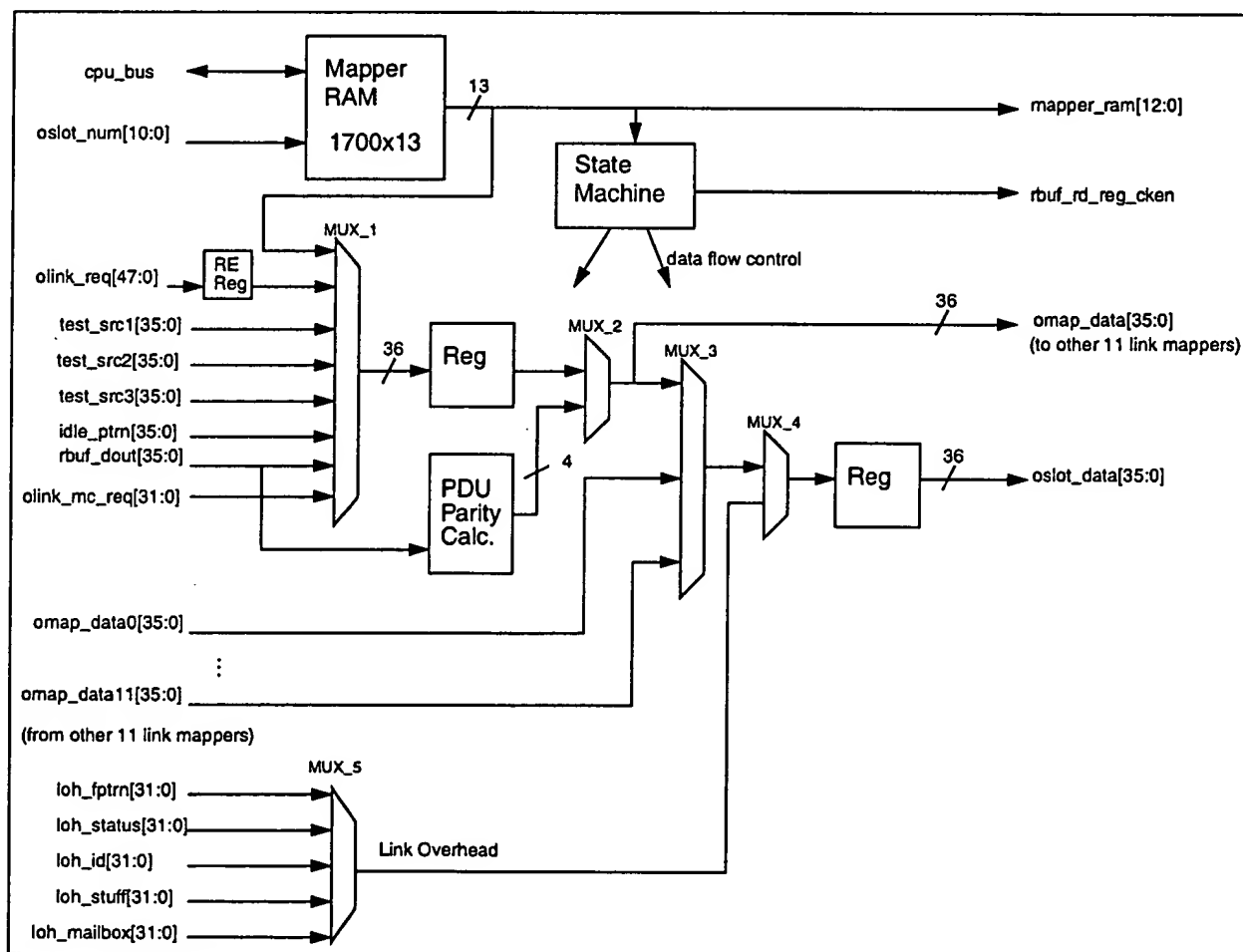


Figure 4-19: Output Link Mapper

Implementation Notes -

Loop back omap_data output to omap_data input so have all 12 links on omap_data in, this will make programming for TDM multicast in Mapper ram easier (will have a 12:1 mux vs an 11:1 mux).

Egress Traffic Timing -

Figure 4-20 illustrates the timing for outgoing data slots. The iTSE implementation will require cclk to be at least

Proprietary and Confidential Information of Onex Communications Corporation

twice the outgoing slot rate. This will insure that the iTSE will have a minimum of two cclk cycles to process each outgoing slot of data. The two cclk cycles per slot are split into two phases in order to identify which cclk edge the oslot_num and oslot_data signals are changing.

The iTSE will allow cclk to be asynchronous to the serial link clocks, the only requirement is that cclk be chosen such that there are always a minimum of two cclk cycles per slot. Because cclk may be asynchronous to the outgoing link, there may more that 2 cclk cycles for any given output slot. In this case both oslot_phase0 and oslot_phase1 will both be deasserted for all but the first two clock cycles for each input data slot. This case is shown during oslot_num = 2 in Figure 4-20. Whenever an extra timing adjust clock cycle is inserted, the signal oslot_phasez will be asserted.

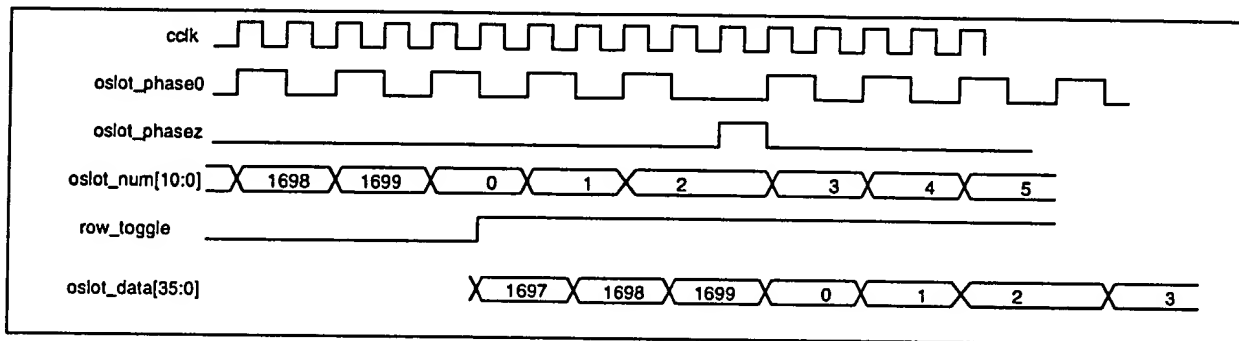


Figure 4-20: Egress Traffic Timing

4.3.1.5.1 Mapper RAM

RESERVED.

4.3.1.5.2 Output Link Request Element

When Mapper RAM indicates it is time to transmit a request element, the highest priority 52-bit request element is fetched from the arbitration module. Since only 36-bits of a RE may be transmitted in a single link slot, it will be necessary to add buffering within the Output Link Mapper module which will buffer the extra bits which cannot be sent in the current slot.

A 3-slot structure will be defined which will be used to transmit 2 52-bit REs and their associated 2-bit BIP2 parity. The contents of the 3 slot structure is shown in the Req Element Slot RAM structure in Figure 4-21.

The timing for fetching REs is shown below. In this example, the 3 slot RE structure is programmed to be transmitted on output link slots N, N+1, and N+2.

The Output Link Mapper module will be responsible for prepending the 2-bit BIP2 parity in bit positions 53 and 52 to create a 54-bit request element.

Proprietary and Confidential Information of Onex Communications Corporation

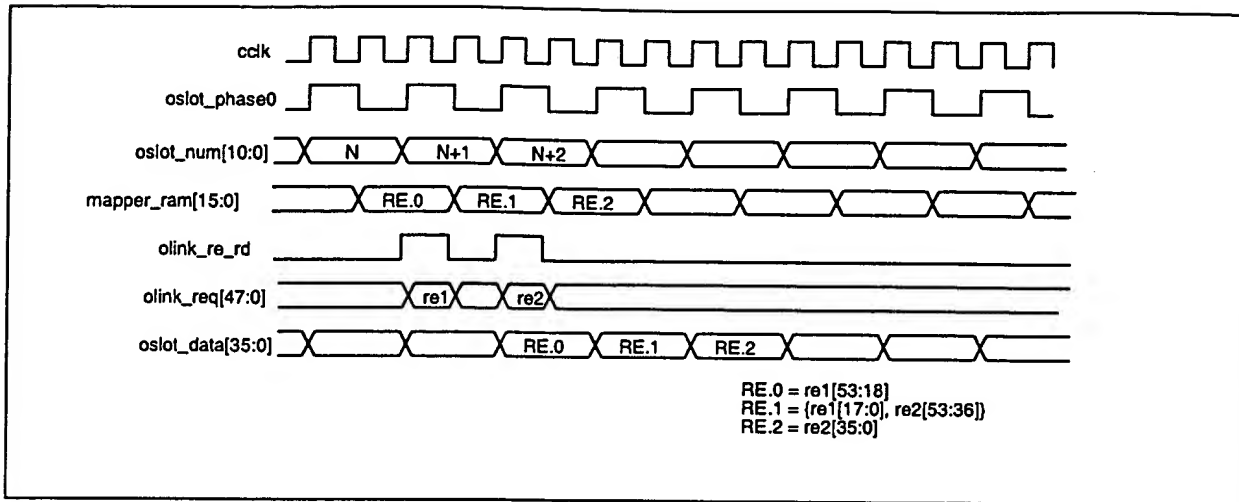


Figure 4-21: Request Element Timing

As we can see from this timing diagram, the RE (olink_req) must be available during the same cclk cycle as olink_re_rd is asserted. This means the Request Arbiter module will need to implement a "pre-fetch" mechanism for the outgoing request elements.

4.3.1.5.3 Output Link Overhead

There will normally be 20 slots used for Link Overhead (LOH). The mapper module will be responsible for inserting contents of these 20 LOH slots into the link data stream. There are 4 types of data which may be inserted into the LOH slots:

LOH Framing Pattern -

This will be a 36-bit value which is common to all output links. It will be Configurable via a software programmable register. This pattern will be used in only 1 of the 20 LOH slots.

LOH Status -

This 32 bit status field will contain only a single bit of status information. In bit 24 the synchronization status of the Grant channel for this link will be carried. All other bits will be fixed at 0.

Note: the 4 tag bits are fixed to all 1's.

LOH Identifier -

This 32-bit will contain an identifier for this switch & link. The field is made up as:

- loh_id[3:0] = link number that the output mapper is instantiated as.
- loh_id[27:4] = iTSE ID number which is SW configurable (via switch_id register in the RISC core).
- loh_id[31:28] = stage number the iTSE is programmed as.

Note: the 4 tag bits are fixed to all 1's.

LOH Stuff -

This 32-bit pattern will be inserted in the LOH slots which aren't used for framing, status, or ID. This pattern will be Configurable via a software programmable register and is common to all output links.

Note: the 4 tag bits are fixed to all 1's

LOH Mailbox -

This 32 bit mailbox will be configurable via a software programmable register. There will be a unique mailbox register for each output link. This mailbox register will provide a mechanism to allow the CPU in this iTSE to communicate with the CPUs attached to its output links. At the time this is being written, there is not any known applications for this mailbox.

Note: the 4 tag bits are fixed to all 1's.

Proprietary and Confidential Information of Onex Communications Corporation

4.3.1.6 Datapath Link CSR

This section summarizes the Control Status Registers (CSRs) used to configure and monitor the operation of an individual Datapath Link module. CSRs include all configurable memory devices within this module, these devices may be individual flip-flops, register arrays or memory arrays.

Note: there is an additional CSR module which contains global control information which is common to all Datapath Link module. This global CSR module is described in Section 4.3.2. Statistics information which is gathered by each individual Datapath Link Module is stored in a centralized Statistics module which is accessed via the global CSR address space.

Unless otherwise noted, the reset value for programmable fields is 0.

Table 4-2: Datapath Link Module CSRs

31	24	23	16	15	8	7	0	Address Offset	
DemapperRam, location 0					all 0's			0x0000	
...					all 0's			...	
DemapperRam, location 1699					all 0's			0x1A8C	
unused address space									
ReceivedPDUCount (don't clear on read)								0x1F80	
TransmittedPDUCount (don't clear on read)								0x1F84	
ErroredPDUCount (don't clear on read)								0x1F88	
ErroredReqCount (don't clear on read)								0x1F8C	
BIP36ErrorCount (don't clear on read)								0x1F90	
PeakBIP36Errors (don't clear on read)								0x1F94	
unused address space									
ReceivedPDUCount (clear on read)								0x1FA0	
TransmittedPDUCount (clear on read)								0x1FA4	
ErroredPDUCount (clear on read)								0x1FA8	
ErroredReqCount (clear on read)								0x1FAC	
BIP36ErrorCount (clear on read)								0x1FB0	
PeakBIP36Errors (clear on read)								0x1FB4	
unused address space									
all 0's			ErrorFlags						0x1FD0
all 0's			ErrorFlagsMask						0x1FD4
0	RBufPduLimit		ILinkDemapperControl		LinkControl		OLinkMapperControl		0x1FD8
all 0's								RxLohSync	0x1FDC
TxLohMailbox									0x1FE0
RxCaptureReg									0x1FE4
unused address space									
0	0	MapperRam, location 0			all 0's			0x2000	
...					all 0's			...	
0	0	MapperRam, location 1699			all 0's			0x3A8C	

Notes on Stats Counter -

Each statistic counter has 2 addresses which it may be read from. One address will automatically clear the counter after the read cycle, the other address will not clear the counter. The counters will saturate at all 1's if the max count value is reached.

ReceivedPDUCount

Cumulative count of the incoming valid PDUs received on this link and forwarded to the Row Buffers. This counter is 20-bits wide.

Proprietary and Confidential Information of Onex Communications Corporation

TransmittedPDUCount

Cumulative count of the outgoing valid (non-idle) PDUs transmitted on this link. This counter is 20-bits wide.

ErroredPDUCount

Cumulative count of the incoming PDUs discarded due to a checksum parity error. This counter is 12-bits wide.

ErroredReqCount

Cumulative count of the incoming REs discarded due to a BIP2 parity error. This counter is 12-bits wide.

BIP36ErrorCount

Cumulative count of the BIP36 errors detected each row time. This counter is 24-bits wide.

PeakBIP36Errors

Maximum BIP36 errors detected in a single row time. This counter is 12-bits wide.

Notes on Error Flags -

Error flags are classified into one of two types: (1) Configuration errors which are errors which are caused by a mis-configuration of the hardware, and (2) Traffic errors which are generated based on the incoming traffic stream. The "Type" column in the error flag description table belows indicates which type of error it is.

ErrorFlags

This 24-bit register contains several flags for error events which may be detected within the Datapath Link Module. Error flags are latched upon detection of the error event and remain latched until they are cleared by software. An error flag is cleared by writing a "1" to that bit position..

Bit	Name	Type	Description	Source Module
0	re_seq_error	Config	Error in RE sequence in Demapper RAM. Sequence which is not RE 0, 1, then 2 has been detected.	Input Link Demapper
1	re_dist_error	Config	Error in RE distribution in the Demapper RAM. This occurs if REs are spaced too close together. Each group of 3 REs need to be spaced at least 16 clk cycles apart.	Input Link Demapper
2	re_parity_error	Traffic	BIP2 parity error detected in a RE. A cumulative count of errored RE's is maintained in the statistics module.	Input Link Demapper
3	pdu_seq_error	Config	Error in PDU sequence in Demapper RAM. Sequence which is not PDU 0, 1, through 15 has been detected.	Input Link Demapper
4	pdu_parity_error	Traffic	Parity error detected in a PDU. A cumulative count of errored PDU's is maintained in the statistics module.	Input Link Demapper
5	slot_parity_error	Traffic	BIP36 slot parity error detected. A cumulative count of the BIP36 errors is maintained in the statistics module.	Input Link Demapper
6	-	Traffic	unused, always read as 0.	Input Link Demapper
7	-	Traffic	unused, always read as 0.	Input Link Demapper
8	tdm_flag_error_0	Config	More that cclks_per_slot incoming TDM slots are valid for this FIFO in a single slot period. This TDM slots which excede cclks_per_slot are lost. cclks_per_slot parameter is configured in the Global CSRs. This TDM FIFO services Row Buffer Addresses 0 through 4 (mod 16).	Row Buffer Mapper
9	tdm_flag_error_1	Config	This TDM FIFO services Row Buffer Addresses 5 through 7 (mod 16).	Row Buffer Mapper
10	tdm_flag_error_2	Config	This TDM FIFO services Row Buffer Addresses 8 through 11 (mod 16).	Row Buffer Mapper
11	tdm_flag_error_3	Config	This TDM FIFO services Row Buffer Addresses 12 through 15 (mod 16).	Row Buffer Mapper
12	tdm_fifo_full_error_0	Config	TDM FIFO overflow. This TDM FIFO services ROW Buffer Address 0 through 4 (mod 16).	Row Buffer Mapper
13	tdm_fifo_full_error_1	Config	This TDM FIFO services Row Buffer Addresses 5 through 7 (mod 16).	Row Buffer Mapper
14	tdm_fifo_full_error_2	Config	This TDM FIFO services Row Buffer Addresses 8 through 11 (mod 16).	Row Buffer Mapper

Proprietary and Confidential Information of Onex Communications Corporation

15	tdm_fifo_full_error_3	Config	This TDM FIFO services Row Buffer Addresses 12 through 15 (mod 16).	Row Buffer Mapper
16	tdm_fifo_ne_error_0	Config	The TDM FIFO is not empty at the end of the row. This TDM FIFO services ROW Buffer Address 0 through 4 (mod 16).	Row Buffer Mapper
17	tdm_fifo_ne_error_1	Config	This TDM FIFO services Row Buffer Addresses 5 through 7 (mod 16).	Row Buffer Mapper
18	tdm_fifo_ne_error_2	Config	This TDM FIFO services Row Buffer Addresses 8 through 11 (mod 16).	Row Buffer Mapper
19	tdm_fifo_ne_error_3	Config	This TDM FIFO services Row Buffer Addresses 12 through 15 (mod 16).	Row Buffer Mapper
20	pdu_limit_error	Config	Set if input PDUs are discarded because pdu_limit has been met. Normally this would be caused by: 1. The request arbiter is configured for more PDUs than pdu_limit. 2. The source device is sending more PDUs than it's been granted. 3. Mis-routed PDUs which are not detected by the PDU parity mechanism.	Row Buffer Mapper
21	pdu_start_error	Config	Set if a PDU start indicator is received for the next half of a PDU while the current half PDU is still being serviced. For example, if input link 0 Demapper RAM is configured for a PDU on slots 0 thru 15 and input link 1 is configured for a PDU on slots 4 thru 19, and both receive PDUs destined for the same output link, this error will be generated.	Row Buffer Mapper
22	-	Config	unused, always read as 0.	Row Buffer Mapper
23	-	Config	unused, always read as 0.	Row Buffer Mapper

ErrorFlagsMask (reset state = 0xFFFFF)

The ErrorFlags are passed through this mask and then logically OR'ed together to generate the dp_link_config_error and dp_link_traffic_error output signals. The mask bit must be a "1" to enable an error flag to be used in asserting the error output signals.

Even if an error flag is masked off, its status can still be read via the ErrorFlags register.

RBufPduLimit

This 7-bit field which specifies the maximum number of PDUs which may be stored in the row buffer in any single row period.

ILinkDemapperControl

This 8-bit register controls the operation of the Input Link Demapper module.

Bit	Name	Description
0	slot_parity_sel	BIP36 parity select, 0 = even parity, 1 = odd parity.
1	re_parity_sel	Request Element BIP2 parity select, 0 = even parity, 1 = odd parity.
2	disable_re_par_check	If set, errored REs are not discarded. Error flag and statistic counters will still be incremented if parity errors are detected.
3	disable_pdu_par_check	If set, errored PDUs are not discarded. Error flag and statistic counters will still be incremented if parity errors are detected.
4	disable_rcv_tdm	If set, all TDM traffic received on this input link is ignored, regardless of the state of the Demapper RAM.
5	disable_rcv_pdu	If set, all PDU traffic received on this input link is ignored, regardless of the state of the Demapper RAM.
6	disable_rcv_re	If set, all RE traffic received on this input link is ignored, regardless of the state of the Demapper RAM.
7	-	

Proprietary and Confidential Information of Onex Communications Corporation

OLinkMapperControl

This 8-bit register controls the operation of the Output Link Mapper module.

Bit	Name	Description
0	slot_parity_sel	BIP36 parity select, 0 = even parity, 1 = odd parity.
1	re_parity_sel	Request Element BIP2 parity select, 0 = even parity, 1 = odd parity.
2		
3		
4	disable_xmt_test	If set, the Idle Pattern will be sent on all Test slots.
5	disable_xmt_tdm	If set, the Idle Pattern will be sent on all output TDM slots.
6	disable_xmt_pdu	If set, the Idle Pattern will be sent on all output PDU slots.
7	disable_xmt_re	If set, the Idle Pattern will be sent on all output RE slots.

LinkControl

This 8-bit register supplies miscellaneous control bits for other modules within the Datapath Link.

Bits	Name	Description
0		
1		
2		
3		
4		
5		
6		
7		

TxLohMailbox

32-bit field which may be inserted into the transmit link overhead slots. The Output Link Mapper RAM must be configured to insert this field at the appropriate slot time.

RxCaptureReg (read only)

32-bit capture register. Any slot on the incoming may be captured and read by software via this address. The Input Link Demapper RAM must be programmed in the appropriate slot time to capture the slot data. Only one slot should be captured in any row time since the capture register will be overwritten for each slot which is programmed via the Demapper RAM for the capture register.

RxLohSync (read only)

Bits 28:24 of the received Link Overhead Status slot may be read from this location. This location is updated for each slot which the Input Link Demapper RAM defines as a LOH Status slot.

Bits	Name	Description
0	gnt_sync	Bit 24 of LOH status. This bit indicates the synchronization status of the Grant channel input of the device at the other end of this link.
1	sp_sync	Bit 25 of LOH status. This bit indicates the synchronization status of the Service Process device at the other end of this link. If the other end of this link is another ITSE, then this bit will always be 0.
2	-	Bit 26 of LOH status. Reserved for future use.
3	-	Bit 27 of LOH status. Reserved for future use.

The rx_loh_sync_mask in the DpGlobalControl register defines which RxLohSync bits are monitored for generating the RxMsgIrq. Even if a bit is masked off, its status can still be read via this RxLohSync register.

4.3.2 Datapath Control

This module implements functions which are either common to or shared by all 12 Datapath Link modules. These functions include:

- CPU bus address space decoding to support the CSRs for each link and the global CSR.
- Global Control/Status registers.
- Timing control memory which will allow for simultaneous support for two different link configurations (i.e., how each of the input link row slots are utilized). This will allow a single ITSE to be used in up to 2 stages of the switch fabric (folded network).
- Logically OR the error_flag from each of the 12 Datapath Link modules and output a single datapath_error_flag output signal.
- Statistics gathering of PDU traffic received in each row buffer.

4.3.2.1 Datapath Memory Map

The memory map for the entire Datapath module is as follows (the address offset is the address offset of that module from the base address of the Datapath module):

- Datapath Link #0 CSR - Address offset: 0x00000, size: 16Kbytes.
- Datapath Link #1 CSR - Address offset: 0x04000, size: 16Kbytes.
- Datapath Link #2 CSR - Address offset: 0x08000, size: 16Kbytes.
- Datapath Link #3 CSR - Address offset: 0x0C000, size: 16Kbytes.
- Datapath Link #4 CSR - Address offset: 0x10000, size: 16Kbytes.
- Datapath Link #5 CSR - Address offset: 0x14000, size: 16Kbytes.
- Datapath Link #6 CSR - Address offset: 0x18000, size: 16Kbytes.
- Datapath Link #7 CSR - Address offset: 0x1C000, size: 16Kbytes.
- Datapath Link #8 CSR - Address offset: 0x20000, size: 16Kbytes.
- Datapath Link #9 CSR - Address offset: 0x24000, size: 16Kbytes.
- Datapath Link #10 CSR - Address offset: 0x28000, size: 16Kbytes.
- Datapath Link #11 CSR - Address offset: 0x2C000, size: 16Kbytes.
- Datapath Global CSR - Address offset: 0x30000, size: 4Kbytes.

4.3.2.2 Datapath Global CSR

This section summarizes the Control Status Registers (CSRs) used to configure and monitor the operation of all Datapath Link module. These CSRs are for control which is common or shared by all Datapath Link modules.

Note: there is an additional CSR module which contains link control information which is instantiated in each Datapath Link module. This link CSR module is described in Section 4.3.1.6.

Unlike the CSR module inside the Datapath Link, the data bus interface to this module is 32-bits wide.

Unless otherwise noted, the reset value for programmable fields is 0.

Table 4-3: Datapath Global CSRs

31	24	23	16	15	8	7	0	Address Offset	
unused								IdlePtrn[35:32]	0x0FB0
IdlePtrn[31:0]									0x0FB4
DpGlobalControl									0x0FB8
LohFptrn[31:0]									0x0FBC
LohStuff[31:0]									0x0FC0
all 0's	LinkTrafficErrorFlags				all 0's	LinkConfigErrorFlags		0x0FC4	
all 0's	LinkTrafficErrorFlagsMask				all 0's	LinkConfigErrorFlagsMask		0x0FC8	
all 0's	LinkMailboxMsgIrq				all 0's	LinkSyncMsgIrq		0x0FCC	
all 0's	LinkMailboxMsgIrqMask				all 0's	LinkSyncMsgIrqMask		0x0FD0	
all 0's						ConfigErrorIsIslotNum		0x0FD4	

Proprietary and Confidential Information of Onex Communications Corporation

PDUcountLink 0..11

32-bit count of the number of PDUs written to each links Row Buffer. This counter is cleared on read. Note: if address minus 0x10 is used as the read address, the count value can be read without clearing the counter.

IdlePtrn

Idle Pattern, 36-bit pattern which is inserted into unused transmit slots. Insertion is controlled by the Output Link Mapper RAM.

DpGlobalControl

Datapath Global Control. 32-bit register which provides control settings common to all 12 Datapath Links.

Bits	Name	Description	Reset Value
3:0	req_valid_time	Number of cclks minus 1 that the request element must be valid when being presented to the arbitration logic. This should never be modified.	0x7
7:4	cclks_per_slot	Minimum number of cclks per slot. This should never be modified.	0x2
11:8	rx_loh_sync_mask	This mask defines which RxLohSync bits will be monitored for generating the LinkSyncMsg interrupt request from each Datapath Link module. The interrupt is generated whenever any unmasked bit changes state from the previous row. A "1" will unmask the bit and enable it to be monitored for IRQ generation. This common mask value is used by all 12 Datapath Link modules.	0xF

LohFptrn

Link Framing Pattern. 32-bit field which may be inserted into the transmit link overhead slots. The Output Link Mapper RAM must be configured to insert this field at the appropriate slot time.

LohStuff

Link Stuff Pattern. 32-bit field which may be inserted into the transmit link overhead slots. The Output Link Mapper RAM must be configured to insert this field at the appropriate slot time.

LinkConfigErrorFlags (read only)

The 12 config error flag outputs of each Datapath Link module may be read from this location. The error flags can only be cleared by writing to the ErrorFlags CSR in each Datapath Link module.

LinkConfigErrorFlagsMask (reset state = 0xFFFFFFFF)

The 12 config error flags are passed through this mask and then logically OR'ed together to generate the Config Error IRQ. The mask bit must be a "1" to enable an error flag to be used in asserting the IRQ. Even if an error flag is masked off, its status can still be read via the LinkConfigErrorFlags register.

LinkTrafficErrorFlags (read only)

The 12 traffic error flag outputs of each Datapath Link module may be read from this location. The error flags can only be cleared by writing to the ErrorFlags CSR in each Datapath Link module.

LinkTrafficErrorFlagsMask (reset state = 0xFFFFFFFF)

Operates identically to LinkConfigErrorFlagsMask.

LinkMailboxMsgIrq

The 12 LinkMailboxMsg IRQ outputs of each Datapath Link module may be read from this location. The IRQ events are latched upon detection and remain latched until they are cleared by software. An IRQ is cleared by writing a "1" to that bit position.

LinkMailboxMsgIrqMask (reset state = 0xFFFFFFFF)

Operates identically to LinkConfigErrorFlagsMask.

Proprietary and Confidential Information of Onex Communications Corporation

LinkSyncMsgIrq

The 12 LinkSyncMsg IRQ outputs of each Datapath Link module may be read from this location. The IRQ events are latched upon detection and remain latched until they are cleared by software. An IRQ is cleared by writing a "1" to that bit position.

LinkMailboxMsgIrqMask (reset state = 0xFFFFFFFF)

Operates identically to LinkConfigErrorFlagsMask.

ConfigErrorSlotNum

When a LinkConfigError is generated, the current input link slot number is latched. The latched value is automatically cleared when it is read. Only when this latch is all 0's will the slot number be latched upon a LinkConfigError. This means only the first LinkConfigError which occurs during a row time will be captured. This is intended to help the wayward software engineer isolate where in the row the configuration error is.

Proprietary and Confidential Information of Onex Communications Corporation

5 Data Multicast Description

This section will describe how the data multicasting & broadcasting is implemented within an ITSE. This section covers the multicasting & broadcasting of data PDUs only, the concept of multicasting TDM traffic is covered elsewhere in this specification. Note: in this section the terms "PDU" and "packet" are synonymous. For the ITSE, "broadcast" can be viewed as simply another form multicast (with a lot of destinations). Therefore, this section will only speak of "multicast" but the reader should realize the this will also be how broadcast is implemented.

5.1 Design Objectives

- Support a one-to-many multicast scheme. Many-to-many multicasting will not be explicitly supported but may be possible depending on the complete system architecture.

Routing tables at the oTPP which will contain the new VPI/VCI for the outgoing cell. This table will be indexed by the *Multicast Flow Identifier*.

5.2 Multicast Operation

In order to support multicast operation in switch fabrics which could be as small as 12x12 ports to as large as 1728x1728 ports, the ITSE will support several methods of multicast operation:

- multicast mask
- broadcast
- multicast ID
- ITTP-based multicast

The multicast controller within the ITSE implements a store and forward mechanism. This means that all PDUs which may be multicast from a given ITSE will be written to a buffer within the multicast controller (MC-CTRL), from there the ITSE will multicast the packet to various destinations. Note: the normal Req-Grant arbitration mechanism is used to get link bandwidth for transmitting the MC PDUs.

5.2.1 Multicast Modes

The multicast modes of operation will be briefly described here. The next section will provide some examples of multicast operation which may make understanding these operating modes a bit easier.

Depending on the size of the switch fabric, several of these multicast operating modes may be used in conjunction to provide an effective multicast solution.

Multicast Mask

5.2.2 Multicast Examples

The ITSE is designed to support a multi-stage packet duplication architecture as shown in Figure 5-1. In this example architecture, the input ITTP will unicast the multicast packets to a multicast controller in stage 1 (switch chip 1.0 in this figure), from there the multicast packet will be duplicated and sent to the multicast controllers in the third stage. The third stage multicast controllers will again duplicate the packet for each output link it is destined for.

The multicast controller implements a store and forward mechanism. This will be explained by using the multicast example in Figure 5-1. In this scenario multicasting will operate as follows:

1. The ITTP will send the multicast packet to the multicast controller (MC-CTRL) in switch 1.0. The MC-CTRL will have buffers in which it will store the received multicast (MC) packet.
2. The MC-CTRL will then duplicate the packet 2 times so that it will now have a total of 3 copies

of the packet. Then for each packet it will replace the routing tag and MC forwarding tags (see the MC PDU format shown in TBD). The packets destined for switch 3.0 and 3.2 will be given the paths to those switch chips. The packet that is being routed through switch 3.1 doesn't need to be multicast from that switch chip, therefore it will be given the path to send it directly to the output iTPP.

3. The MC-CTLR will then forward the 3 MC packets to the 3 destinations (switches 3.0 and 3.2, and the iTPP off switch 3.1). This forwarding will again use the normal Req-Grant arbitration mechanism for obtaining bandwidth.
4. At switch chips 3.0 and 3.2, the MC packets will be duplicated and sent out on output ports which are specified in the MC copy field of the MC packet. Since this is the last multicast stage, the MC packets do not need to be modified prior to transmission. The identical MC packet is sent on all the appropriate output links. Of course, the normal Req-Grant arbitration mechanism will be used to get link bandwidth from the last MC stage to the iTPP.

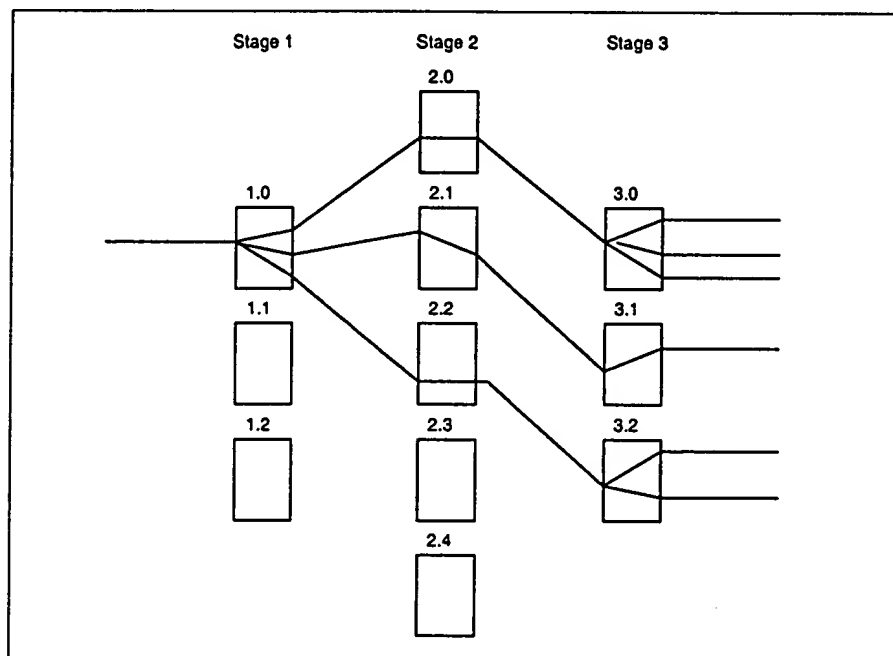


Figure 5-1: Multicasting within the iTAP Switch Fabric

For the MC-CTLR at the first stage (switch 1.0 in Figure 5-1), we see that it will need to modify the routing and MC copy tags within the MC packet prior to transmitting the duplicated packets. This means that the MC-CTLR will need to have information which is specific to each MC flow which may use the MC-CTLR as a copy forwarding stage. The following parameters will need to be stored in the iTSE MC-CTLR prior to the arrival of the MC packet:

Per MC Flow -

- 11 bits to identify the source iTPP of the MC packet.
- 14 bits to identify the MC Flow ID as assigned in the sourcing iTPP.
- 4 bits to identify the cache entry to use (this is discussed later).

Per output destination to which the packet will be multicast to -

These bits will be used to replace the fields currently in original multicast PDU.

- 28 bits for routing tag.
- 12 bits for MCcopy tag.
- 1 bit for the new broadcast field.
- 1 valid bit which defines whether this is a valid entry, if clear then ignore this entry.

Proprietary and Confidential Information of Onex Communications Corporation

These bits will be used in the creation of the request element used for link BW arbitration.

- 5 bits for VoqID.
- 3 bits for priority.
- 3 bits for res field.

The RouteTag will be the same one used above when copying the PDU. The ReqId will be a created by the ITSE multicast controller in order to identify the returning grant.

Thus, for each destination which the multicast PDU must be copied to we need a total of 53 bits of information.

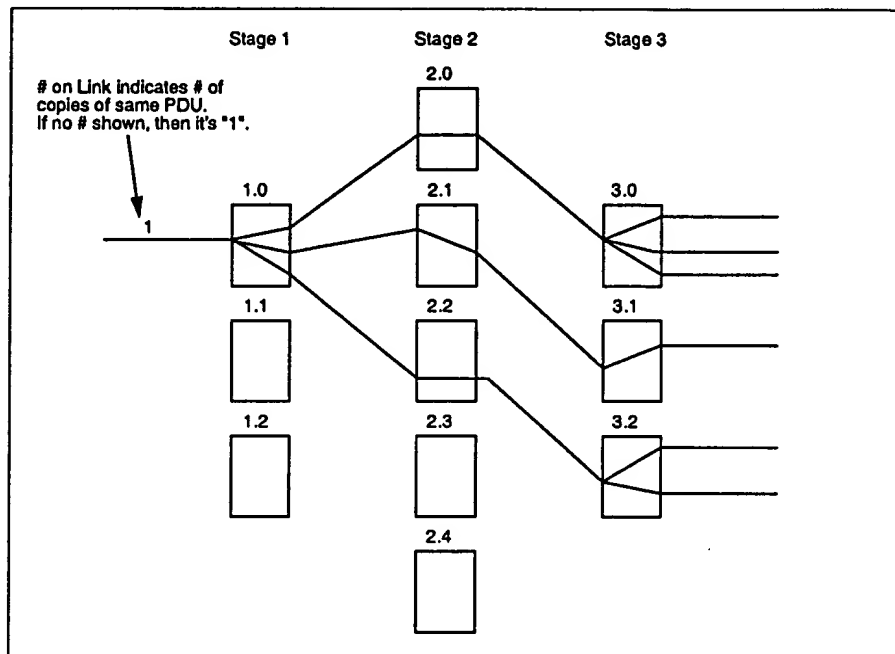


Figure 5-2: Multicasting w/ Multicast ID Mode

5.2.3 Single PDU

The figure below illustrates a typical arbitration and data passing sequence for a data PDU.

Proprietary and Confidential Information of Onex Communications Corporation

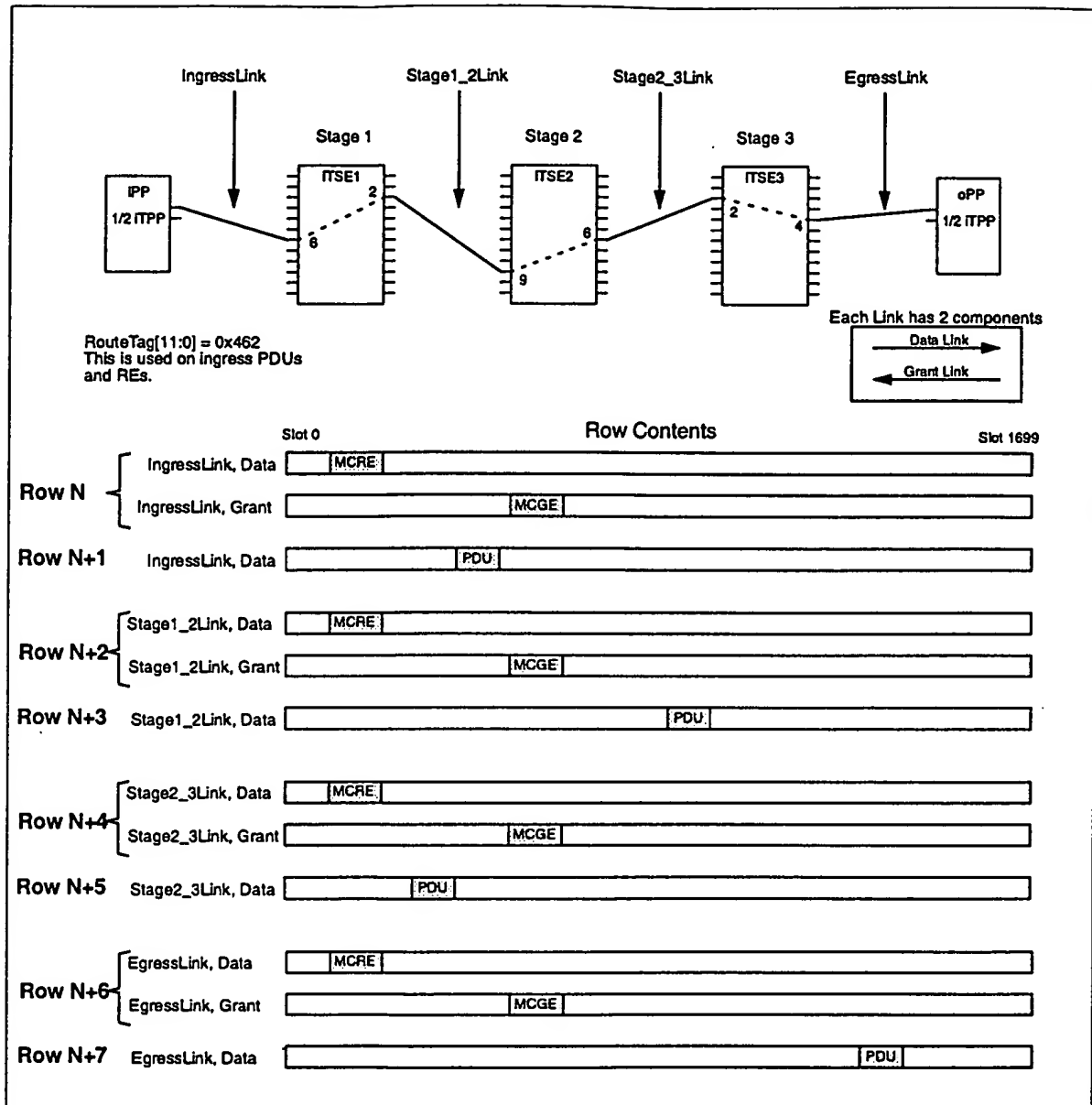


Figure 5-3: Single MC PDU Through a 3-Stage Switch Example

5.3 Multicast PDU Structures

5.3.1 Multicast Data PDU Format

The format for the 16-slot multicast data PDU is shown below. The shaded fields are identical to the fields in the unicast data PDU format are described in Section 2.2.4.1..

Table 5-1: Multicast Data PDU Format

Slot	35	31	24 23				16 15				8 7		0
0	Parity	PDU	BC	MC	RouteTag								
1	Parity	Ver	ValidPBytes			VOQID		FragId	A	McCopy			SeqNum
2	Parity	McCacheTag	-	-	-	SrcPortId				SrcMcFlowId			
3	Parity	Payload_Byte_0			Payload_Byte_1			Payload_Byte_2			Payload_Byte_3		
...	Parity	payload bytes											
15	Parity	Payload_Byte_48			Payload_Byte_49			Payload_Byte_50			Payload_Byte_51		

Note: reserved bit positions are indicated with a "-". The default state for these bits is 0.

BC - Broadcast

This bit will be set if this is a broadcast PDU.

MC - Multicast

This bit will be set if this is a multicast PDU.

McCopy

12-bit Multicast Copy field. This field identifies which output link this multicast PDU must be multicast to. McCopy[0] is for output link #0, McCopy[1] is for output link #1, etc. If all 12 bits are cleared and the MC field = "01", then the multicast controller will use the multicast cache and perform packet duplication based on the contents of the cache.

SeqNum

SeqNum is the fragment sequence count, it will be incremented for each fragment. The SeqNum will start at 0 for the first fragment. If there are more than 16 fragments to the PDU, this SeqNum will roll over past 15 and continue counting.

McCacheTag

4-bit tag which identifies the multicast cache entry to use when duplicating this multicast PDU.

SrcPortId

This 11-bit field identifies the source port processor for this multicast PDU.

SrcMcFlowId

This 14-bit field identifies which multicast flow this PDU is associated with. This is the flow ID from the input Port Processor's multicast ID space.

5.3.1.1 Background on Multicast Data PDU Fields

This sections provides a bit of background on why the fields which are present in the multicast data PDU defined in Table 5-1 amd how they are used.

BC & MC bits -

The usage of these bits will depend on who is processing the packet.

Proprietary and Confidential Information of Onex Communications Corporation

For the iTSE multicast controller the algorithm shown in Figure 5-4 is used.

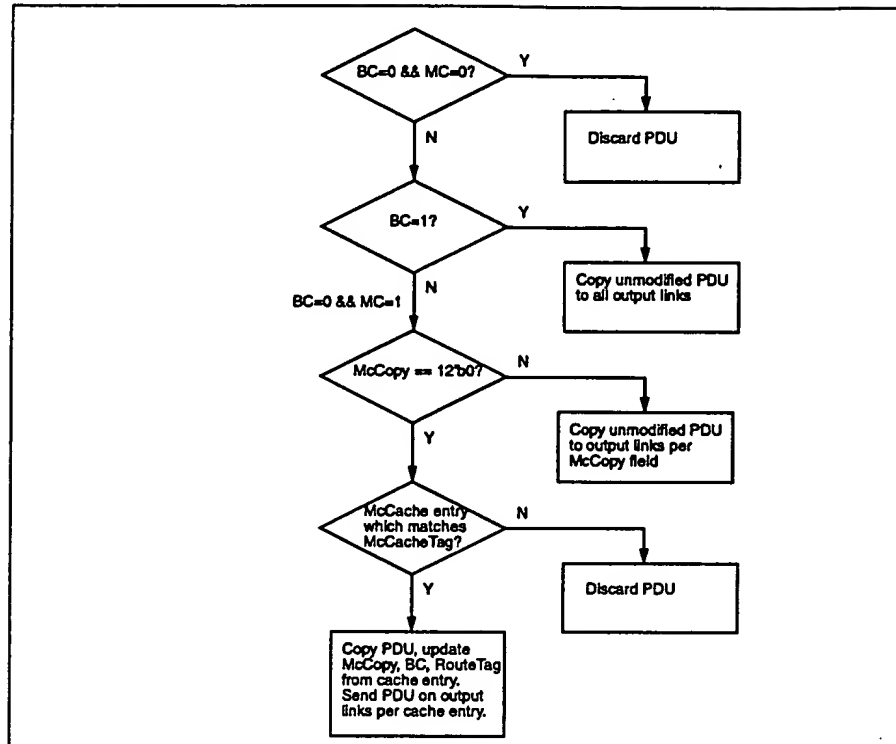


Figure 5-4: iTSE MC Controller Processing of BC & MC Bits

For the output Port Processor the algorithm shown in Figure 5-5 is used. This algorithm assume the ITPP already knows that it is dealing with a multicast packet. The current plan is that all multicast/broadcast PDUs will be sent a virtual output queue which is reserved for MC/BC packets.

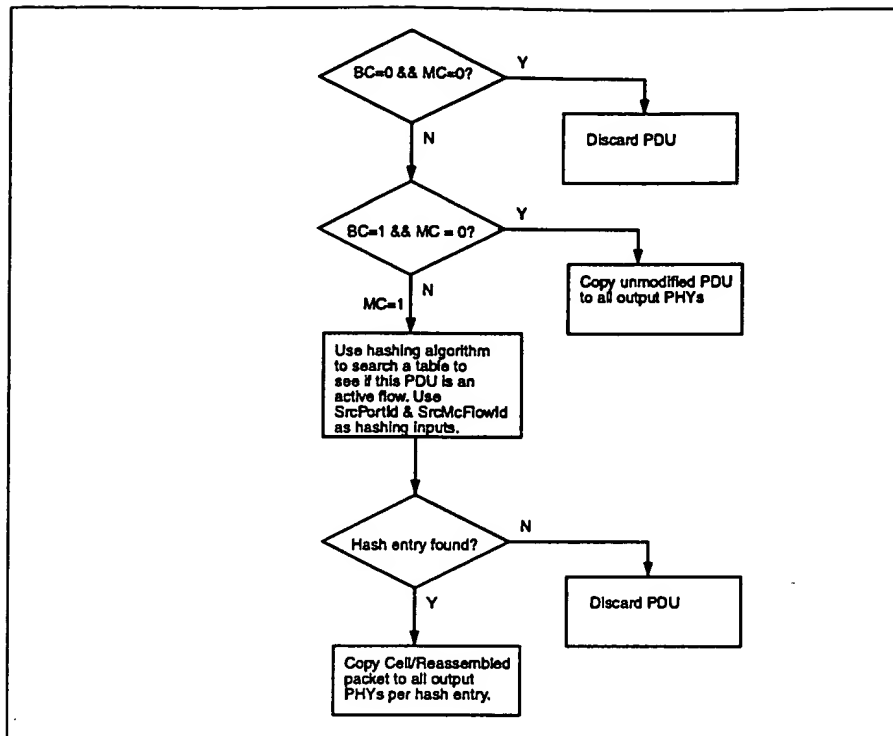


Figure 5-5: ITPP Processing of BC & MC Bits

McCopy -

This field is only used by the iTSE multicast controller. It is used to determine which output links the PDU should be copied to. See the flowchart in Figure 5-4 on how this field is used.

The port processor will always ignore this field.

McCacheTag -

Because we're implementing a store and forward technique within the iTSE multicast controller, we'll need some information (more than what can fit in the MC PDU header) which determines where to forward the MC packet to and the new parameters which need to be replaced for each forwarded copy. This information will be stored in a cache within the iTSE. The cache entries are loaded by having the input ITPP forward a "MC Parameter" PDU prior to each MC data PDU. Since a PDU is only 64 bytes the MC Parameter PDU will only be able to carry enough information to allow the MC data PDU to be copied to 7 unique destinations.

Since 7 destinations may not be enough for a multicast flow, a mechanism is needed to allow for more destinations. This mechanism will be the "Cache Tag" which will allow a single MC flow to have multiple valid cache entries.

Proprietary and Confidential Information of Onex Communications Corporation

5.3.2 Multicast Parameter PDU Format

The format for the 16-slot multicast parameter PDU is shown below.

Table 5-2: Multicast Parameter PDU Format

Slot	35	31	24	23	16	15	8	7	0							
0	Parity	1	1	0	0	RouteTag										
1	Parity	McCacheTag			-	-	-	SrcPortId		SrcMcFlowId						
2	Parity	V	-	BC	-	RouteTag										
3	Parity	-	-	-	VoqId		-		res	Priority	McCopy		-	-	-	-

Slots (addresses) 2 & 3 contains the cache entry parameters for 1 destination. Six more destinations may be defined in the same manner using slots 4 thru 15.

Note: reserved bit positions are indicated with a "-". The default state for these bits is 0.

Slot 0 -

This word is used to route the Multicast Parameter PDU from the input iTPP to the iTSE multicast controller. It's fields are those for a unicast packet as defined in Table 2-3.

Slot 1 -

This word identifies the MC session parameters for this cache entry. The fieds will match those from the multicast data PDU which will be following this parameter PDU.

Slots 2 & 3 -

Thise two words contain the cache entry parameters for 1 destination. When the MC PDU is duplicated the BC, RouteTag, and McCopy fields will be replaced with those in these slots.

The VoqID, res, and Priority fields are used (along with the RouteTag) for creating the request element which will be used to arbitrate for link bandwidth for sending the duplicated data PDU.

Proprietary and Confidential Information of Onex Communications Corporation

5.4 Port Processor's Role in Multicasting

5.4.1 Input Port Processor

5.4.2 Output Port Processor

The output fTPP will receive multicast packets from the switch fabric. For each MC packet it must determine (1) if the MC packet is a member of one of the potentially 16K active outgoing MC flows which the fTPP can support and (2) which Utopia PHYs the packet must be multicast to.

In order to determine if the MC packet is a member of an active MC flow, the fTPP will need to perform a hashing algorithm on the fields from the MC PDU which identify the MC flow. These fields are the SourcePort

The Port Processor board may potentially be supporting up to 192Is designed with multiple PHY ports attached to the fTPP's Utopia bus.

5.4.3 Multicast PDU Latency

Since the multicasting mechanism uses a store and forward approach, it will take longer to get multicast packets through the switch fabric. In the example of Figure 5-1, the best case time for sending a MC packet through this 3 stage switch fabric is calculated as follows:

1. 2 row times are need to get the MC packet from the fTPP to the

Proprietary and Confidential Information of Onex Communications Corporation

6 Multicast Implementation

Possible 2nd Tensilica which is used to handle the multicasting is here.

7 Link Bandwidth Arbitration

7.1 Theory of Operations

7.1.1 Overview

Each port processor operates without any knowledge of what the other port processors are doing. As a result, when they go to send their PDUs, they need to know 2 things:

- Does the output port processor have room in its queues for this PDU?
- Is there bandwidth in the chosen path to get the data from one end of the switch fabric to the other without packet loss?

The arbitration mechanism will check both of these two criteria and send back a grant to the requesting port processor on a PDU by PDU basis. When a port processor has been given a grant it knows for certain that the data will make it to the output port processor (barring system failure).

Each row time, the port processors will make a request for each group that it wishes to send data on in the next row. This request will be a message which is broken into 96 request elements, one element for each possible data group requested. These request elements will be multiplexed in with the data stream (see Overview & Datapath chapters). The requests stream through the switch and are 'knocked out' based on a priority field. Since the 12 inputs could all converge on a single output, the outgoing link will not be able to handle the traffic presented to it. The highest priority traffic should be allowed to go through the switch fabric. A small buffer pool exists in each output link to hold some of the requests when multiple requests come into the switch chip which are destined for the same output link. At the far end of the switch fabric, the port processor will make a decision to grant or deny a request based on its QOS queues. The port processor will then source a grant message which also travels through the switch fabric, but in an *out-of-band* overlay network which goes in the opposite direction of the switch fabric. The grants will be written without regard to priority into a fifo and read in order of arrival time.

7.1.2 Basic Algorithm

The arbitration mechanism will work as follows:

1. At the start of a row time the input port processor will begin outputting its request message, made from a stream of request elements. A request element is a request for a single group's worth of bandwidth in the switch fabric destined for a particular port processor. The format of the request elements are shown below.
2. The first stage in the switch fabric will look at each request from all 12 input links as well as the multicast and control message controller. The requests traverse the switch fabric by using a self routing tag which indicate the hop-by-hop output ports used at each stage of the switch fabric. At this time, the Stage 1 hop-by-hop field will be replaced with the input port number that the request entered on. This parser logic will be able to handle all the requests from all 14 request sources within a single request element time.
3. The requests for each output link will be stored in a buffer pool. As long as buffers are free, requests will be stored. As soon as there are no free buffers, lower priority requests will be overwritten with higher priority ones. The request buffers will be able to support 12 input links all converging on a single output, meaning that 12 request elements can be written to the buffer pool every 'request' time. Requests are evicted from the buffer pool based on priority and age. The youngest lowest priority requests will be dropped, and the highest priority oldest requests will be kept.
4. After a programmable amount of time, the request buffers will be read from by the switch Mapper. After a request is read the request element deleted from the buffer, making room for another request element. The output Mapper will only read 96 request elements- it will not over-request an output link. Any requests still in the buffers will be dropped.
5. This happens all the way to the end of the switch fabric and into the port processor. The port processor will make a decision to accept or reject the request based on the QOS field. Then, it will source a grant message. The grant message uses the modified self routing tag of the request element to traverse the switch fabric backwards using an overlay network.

6. The grant path in the switch uses another instantiation of the parser logic and a set of buffer fifos which get written to and read out of based on arrival time. (The links will be scanned in order and written into the fifos in that same order- link0 to link 11. In addition a mapper and demapper will be used to determine where in the link the grants should be placed. It is the intent that they are all adjacent to each other in the row.

7.1.3 Folded switch fabrics

RESERVED

7.1.4 Multicast Support

In addition to the the 12 input links, provisions need to be made for multicast traffic as well as request messages made by the local processor. Multicast request elements that flow into a switch will flow through the switch fabric the same as standard unicast request elements. At the point where the message needs to be multicast the hop-by-hop field's bit code for that switch stage will indicate that the request is multicast. The request will be forwarded to the multicast controller. On the grant path, the multicast controller will simply source a grant if there is room for the data in the multicast recirculating buffers. Once the data has been transmitted to the multicast buffer, the multicast controller will examine the data header and determine which output links it needs to be sent out on. At this point, it will source a number of request messages which will look to the request-controller as if the switch had 13 inputs, not 12. They will be handled the same as unicast requests from one of the input links.

The arbitration algorithm has been designed with the intents that each request message will request a single group. To aid the multicast manager, however, there is a bit in the request/grant element which indicates if the request is for 1 or 2 groups.

7.1.5 Arbitration Message Element Format

Below are the actual bit patterns of the request and grant elements. Each of these request elements requests for a single data group on the switch link. It takes 96 of these to request an entire ITAP row. Serialized versions of these stream between switch elements.

The request/grant messages will take the form of a self routing message with a 3 bit priority, 7 bit sequence number and either a 5 bit Virtual Output Queue ID or congestion indicators. The Request Element has 2 forms- one is for a 7 stage network, the other is for 5 or smaller stage networks. In these cases the 6 and 7th stage self routing tag is replaced with a QOS field. The switch should simply pass the bits which are set in these fields.

51			48	47			44	43			40	39			36	35			32	31			28	27			24	23							16	15									8	7	6	5	4	3	2	1	0
Stage 1		Stage 2		Stage 3		Stage 4		Stage 5		Stage 6		Stage 7		ReqID				Output Port ID				Num	unused	Priority																													
Stage 1		Stage 2		Stage 3		Stage 4		Stage 5		QOS				ReqID				Output Port ID				Num	unused	Priority																													

Table 7-1: Request Message Element (7 Stage)

51				48	47					44	43					40	39					36	35					32	31					28	27					24	23																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											</
----	--	--	--	----	----	--	--	--	--	----	----	--	--	--	--	----	----	--	--	--	--	----	----	--	--	--	--	----	----	--	--	--	--	----	----	--	--	--	--	----	----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	----

Table 7-2: Grant Message Element (7 Stage)

- Stage1...Stage7: The Stage fields indicate the output link number that the given switch chip should forward the message to. Valid request messages are those with output link numbers between 0 and 11. Other values are reserved for special commands. Each switch chip will need to know at which stage of the network it is located.

Bit Field	Connects to...	Notes
0000	Output Link 0	
0001	Output Link 1	
0010	Output Link 2	
0011	Output Link 3	
0100	Output Link 4	
0101	Output Link 5	
0110	Output Link 6	
0111	Output Link 7	
1000	Output Link 8	
1001	Output Link 9	
1010	Output Link 10	
1011	Output Link 11	
1100	Multicast/Control Mes- sage	
1101	reserved	Spare
1110	reserved	Spare
1111	No Request - IDLE	Need to send something as a place- holder if no requests are pending for this group

Figure 7-3: Stage X Bit Field Codes

- **QOS:** This 8 bit field refers to the queue which the data is destined for in the Output Port Processor. The Output port processor is expected to use this number to determine if it should grant or reject the request.
- **OutputPortID:** This is a 7 bit field used for the output port processor to know which virtual output queue to place the data.
- **ReqID:** As grants stream back to the input port processor which made the requests, the port processor needs to have a way to identify which grant goes with which request. By providing an identification number field, the port processor has a way to quickly associate the grant with a specific request. It is expected that the port processor will insert 0 into the sequence number for request 0 and 1 for request 1, all the way up to 93 for the request 93.
- **Num:** These 2 bits indicates if the request/grant is for 1,2,3 or 4 groups. Although it is expected that the normal port processor data flow will be on a group by group request basis, the multicast controller will operate more efficiently if it has the ability to easily arbitrate for multiple groups. This is needed to simply speed up the passing of the multicast control packets. The coding of these bits is as follows:

Num (0b)	# of Groups
00	1
01	2
10	3
11	4

- **Priority:** This is a 3 bit field to indicate the importance of a request. Highest priority is seven (0b111), lowest priority is zero (0b000). This priority is the priority of the request or grant element through the switch fabric.
- **Res:** These bits will be carried through the switch fabric by the ITAP Switch, the port processor may do what it wants to with these bits for passing additional signaling information across the

switch fabric.

7.1.6 Calculating slot numbers of request element arrivals

The mapper and demapper rams on the data path will have entries in them for the request elements. The equations below are useful for calculating which entries in the rams will be setup for request elements.

These equations assume that the specified 3...8 slot timing is used, and that multiples of 16 slots will be used unbroken between programmable fill time of 4 is specified. 4 refers to the number of request element times to wait before outputting request elements from any switch stage. Since the request elements are sent in pairs, this is an even number. Due to the internal timing of the switch, there is always a fill time of 2- otherwise nothing would have been written into the buffers yet.

There are several equations below. The first equation generates the starting slot number which request elements will come in to a given switch stage at. The second equation generates the slot number which request elements will start leaving a switch stage at. The third equation is used to calculate the slot numbers which the row demapper ram should be programmed for.

Dependent Variables:

- StageNum - the stage number of this switch element. Values shall range from 1 to 7.
- PFT - programmable fill time, an even number which is the number of request elements that the input links are allowed to source before the output link starts up. Values range from 2 (minimum to XX (maximum) and are even numbers. It is expected that 2,4 and 6 are the values used by the switch fabric.
- PLD - Pipeline delay for output mapper. Nominally this is 18 clock cycles ~ 9 slots.
- RET - 11, the number of slots in the request...data...request...data pattern.
- PRET - 32, the number of slots used to offset th
- PPDelay - the number of slots that the port processor waits before sending any requests. This number should be zero.

Once the starting numbers have been calculated, the following equation is used to calculate the slot numbers that the demapper ram should be programmed to for a particular element. Since every request element spans 2 adjacent slots in the chosen timing, the equations below output the first slot number that the request appears on.

7.1.7 Request-Grant Arbitration Cycle Timing

The arbitration cycle timing for varying sized switch fabrics and fill times has been calculated. The following has been assumed in all of the calculations:

- Request Elements are mapped into the row as 3 slots carrying 2 request elements followed by 8 slots of data. This yields 1 request element every 45ns.
- Grant elements are mapped into a 2.2gbps serial stream as 3 slots carrying 2 grant elements.
- The time it takes an output port processor to accept or deny a request element is 125ns.
- The electrical delay between the port processors and switch fabric is set to 500ns. This delay occurs 4 times in 1 round trip.

The Request Elements travel at a rate of 1 every 45 ns, grant elements travel at a rate of 3 per 2 slots, with no padding between groups of 3 at a data rate of 2.2gbps, this is 1 every 25 ns. The grants can be sent faster than they will be received by the switch fabric. As a result, the last grant element timing will be based on the speed of the request message rather than grant speed since the request element is the limiting factor.

For each network, the following times are given:

- **First RE in:** the time that the first request element gets to the output port processor
- **Last RE in:** the time that the last request element in a message gets to the output port processor
- **First GE in:** the time that the first grant element gets back to the input port processor
- **Last GE in:** the time that the last grant element gets back to the input port processor

	Fill Time	4	5	6
--	-----------	---	---	---

Proprietary and Confidential Information of Onex Communications Corporation

Size of Switch Fabric		*All times are in ns		
1 Stage Switch Fabric				
	1st RE In	1253	1298	1343
	Last RE In	5573	5618	5663
	1st GE In	2570	2615	2660
	Last GE In	6890	6935	6980
2 Stage Switch Fabric				
	1st RE In	1433	1523	1613
	Last RE In	5753	5843	5933
	1st GE In	2865	2955	3045
	Last GE In	7185	7275	7365
3 Stage Switch Fabric				
	1st RE In	1613	1748	1883
	Last RE In	5933	6068	6203
	1st GE In	3159	3294	3429
	Last GE In	7480	7614	7749
4 Stage Switch Fabric				
	1st RE In	1793	1973	2423
	Last RE In	6113	6518	6743
	1st GE In	3454	3634	3814
	Last GE In	7774	7954	8134
5 Stage Switch Fabric				
	1st RE In	1973	2198	2423
	Last RE In	6293	6518	6743
	1st GE In	3748	3973	4198
	Last GE In	8068	8293	8518
6 Stage Switch Fabric				
	1st RE In	2153	2198	2423
	Last RE In	6413	6518	6743
	1st GE In	4043	4313	4583
	Last GE In	8363	8633	8903
7 Stage Switch Fabric				
	1st RE In	2333	2648	2963
	Last RE In	6653	6968	7283
	1st GE In	4337	4652	4967
	Last GE In	8657	8972	9288

The following calculations show the travelling of the request messages through the ITAP Switch Fabric. The following delays are encountered when a request element is sent by a port processor.

- **Ted**: Electrical delay. Start of Row is asserted, the entire request message has been assumed to be built already by the scheduler. The time to read the row and the electrical delay associated with getting the signal from the port processor to the switch fabric is 500ns (maximum).
- **Trm**: Request Message Transit Time. The transmission time of a single request element is bound up in the sending of 2 request elements in the 3 slot...8 slot timing. As a result it takes 11 slots to transmit 2 request elements. There will be 48 of these 3...8 timing groups. Therefore the width of

Proprietary and Confidential Information of Onex Communications Corporation

a request message is over 48×11 slots = 528 slots. A slot is transmitted by the ITAP link in 8.17ns, so a request message will be 4.314us wide.

- **Tgm:** Grant Message Transit Time. The transmission time of a single grant element is bound up in the sending of 2 grant elements in 3 slots. The grants can be packed into the row as tightly as possible. Therefore the fastest grant message time is in 96 grants * 1.5slots per grant = 144 slots. Over the grant link, the bandwidth is 2.2gbps, so that a 36 bit slot takes 16.36ns. Therefore the grant transit time is 2.355us. But, since the request message is slower than the grant message, the size of the grant message will be the same as the request message since the request elements feed the grant stream. $Tgm = Trm...$
- **Tpft:** Programmable Fill Time. It is possible to allow the request buffers to start filling up before any requests are sent out. Although this adds time to the round trip timing of the request message, it helps ensure that the highest priority requests are forwarded at each stage of the switch fabric. This fill time would be manifested in the way that the mapper ram is programmed. The slots where the request elements would be programmed would be set deeper into the mapper RAM so that they occurred later in the row. Although this setting could be anything, multiples of the 3-8-3-8 timing are used below as a practical example. The programmable fill time must be an even interger multiple with a minimum value of 2. The calculation from Tpft to slots is simply $11 * 0.5 * Tpft$. In the examples a fill time of 4 has been chosen. This implies that $(8+3) * 2$ slots occur before the request elements are output; 22slots = 180ns.
- **Tpl:** The ITAP Switch chip has an internal request pipeline latency of 18 clock cycles (9 slots) for the request elements. $8.17 * 9 = 73ns$
- **Topp:** Output Port Processing Time. This is given as 32 clock cycles, which @250MHz is 16 slots. 16 slots will be the metric used (which is a full group time). This is 130ns
- **Tgdl:** Grant Delay. 3 Slots needed to extract and switch grant elements, 1 slot to process, and 3 more to assemble the grant elements= 7 slots. $16.36ns * 7 = 114.52ns$.

Below is a picture illustrating an arbitration cycle's request and grant messages flow through a 7 stage switch fabric with the fill time set to 4 request elements.

Factor	Time	Cumulative Time	
IPP to SWF(ted)	0.500us	0.500us	
Switch Pipeline Delay (tpl)	0.073us	0.573us	
7*Fill Time	1.260us	1.833us	
SWF to OPP (ted)	0.500us	2.333us	First Request gets to OPP.
Request Message Width	4.314us	6.647us	Last Request to OPP
OPP Processing Time	0.130us	2.463us	
OPP to SWF (ted)	0.500us	2.963us	
Switch Pipeline Delay	0.073us	3.036us	
7* Grant Delay	0.801us	3.837us	
SWF to IPP (ted)	0.500us	4.337us	First Grant gets to IPP
Request Message Width	4.314us	8.651us	Last Grant gets to OPP

Proprietary and Confidential Information of Onex Communications Corporation

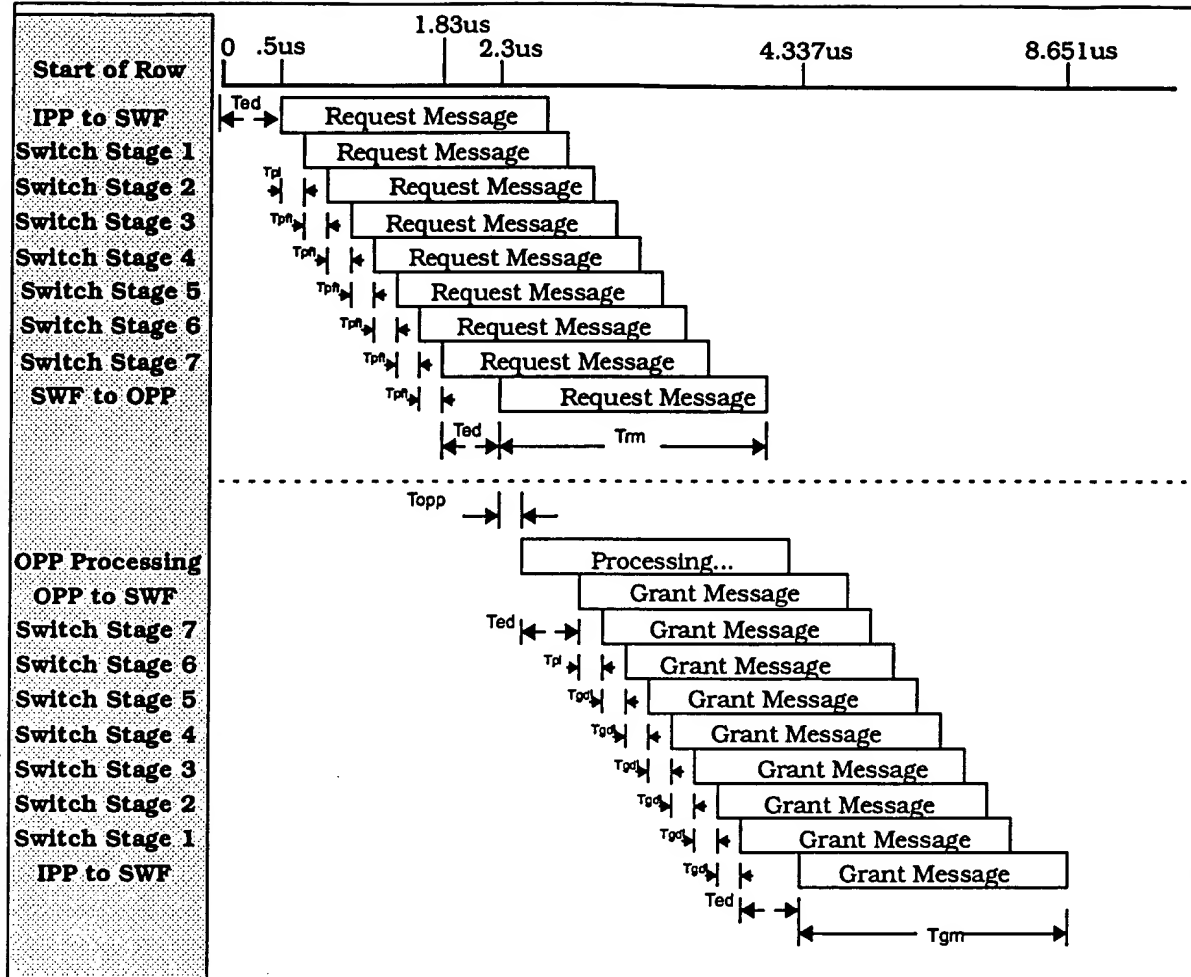


Figure 7-4: Arbitration Cycle In- Flight Timing

7.2 Link Bandwidth Arbitration Implementation

7.2.1 Request Implementation

The request path is implemented in 2 major functional units. These are the request parser and the request arbiter.

Below is a top level block diagram of the arbitration logic. The Request Parser examines the hop-by-hop self routing tag of the request elements and forwards the requests to the appropriate output link request logic. It needs to replace the 'current' stage number field with the input link number which the request came in on. This keeps a record of the reverse path so that the grant can get back to the input port processor. The output link logic will handle 2 requests every clock cycle, looking at the requests and writing them to the request buffer pool. The Output Buffer Logic reads the buffer pool and sends out the highest priority requests. The request parser is instantiated once in the design for the request elements, and the output link buffer logic is instantiated once per output link (12 times total for the request elements).

Proprietary and Confidential Information of Onex Communications Corporation

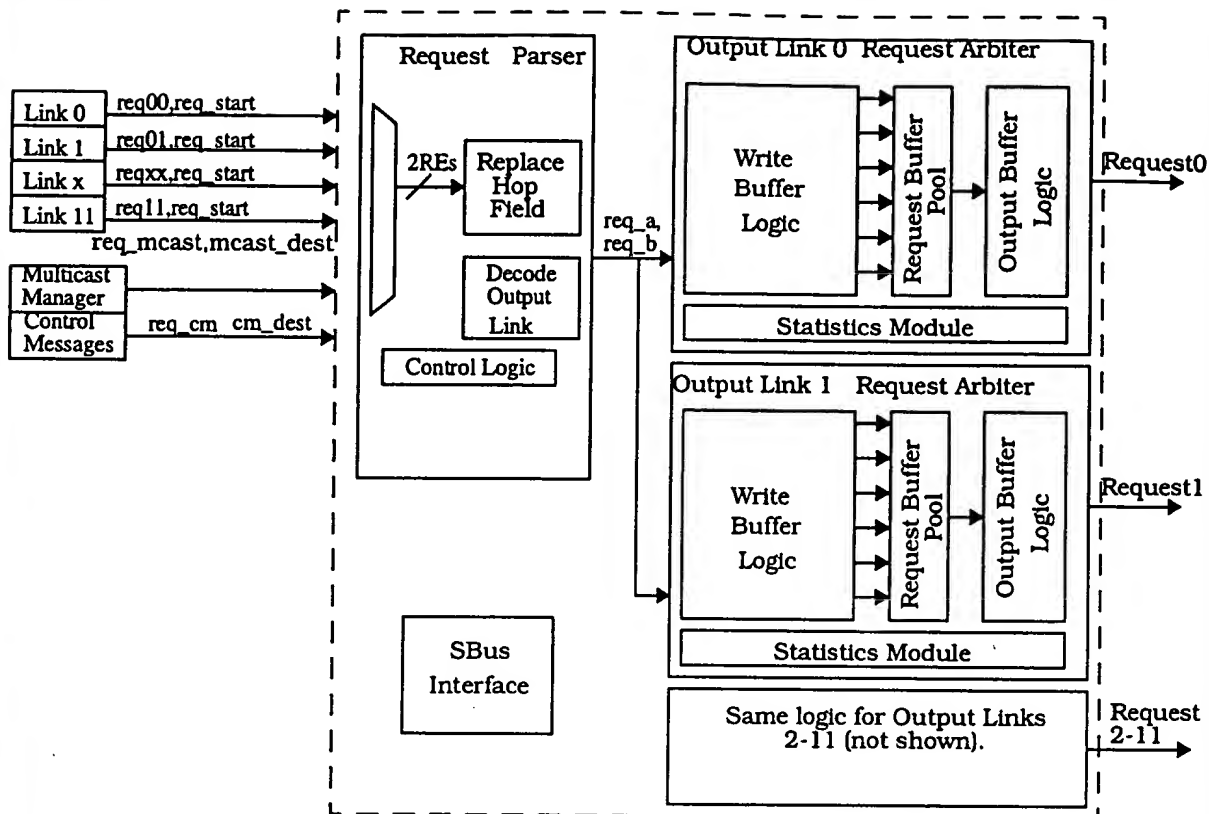


Figure 7-5: Request Arbitration Logic

At the top level the I/O for the request grant path will be the following

Signal Name	Width	I/O	Note
cclk	1	I	Core clock -250MHz
reset_b	1	I	Active low module reset
SBus Ctrl Sigs	X	B	Slave Bus Interface
sor_en	1	I	Start of Row
req_mcast	52	I	Multicast request message.
req_uproc	52	I	Tensilica request message
mcast_link_en	12	I	1 Bit per output link that the req should goto
uproc_link_en	12	I	1 Bit per output link that the req should goto
req_in[11:00]	52	I	Request Element from input links #0-#11
req_start[11:00]	52	I	Request Element Enable from links #0-#11
req_out_en[11:00]	12	I	Pulse to read next request element of output link
req_out[11:00]	52	O	Request for output links #0-#11
req_cm	52	O	Request for inband Control Messages
req_mcast	52	O	Request for the multicast buffers

Table 7-6: Request Arbitration Top Level I/O

7.2.1.1 Request Parser

The request parser is responsible for taking the 12 input links, multicast controller and inband messaging controller request elements and determining which output links they need to goto. The parser needs to have the ability to process all of the request elements every 8 clock cycles (which is the maximum incoming request element rate). Since there are 14 inputs, 2 inputs will be processed every clock cycle with a spare clock cycle in case something can't make timing and needs to be

Proprietary and Confidential Information of Onex Communications Corporation

registered.

The request arbitration mechanism will receive request elements from the datapath. Below is a timing diagram of the interface.

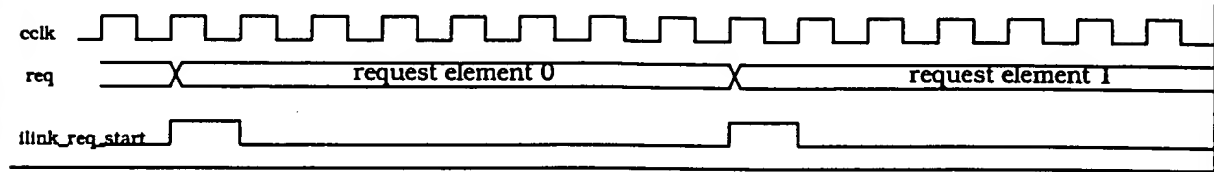


Figure 7-7: Request Parser Input Timing

Upon every ilink_req_start a new request element is ready. The req_start signal from all of the input links will be registered, and every 7 clock cycles will be tested to see if a request element is ready. The fastest that these req_start signals can come in is every 8 clocks, so the request parser is assured of getting to all of the input REs. This interface is duplicated 12 times on the input of the request parser (1 for each input link).

Below is a timing diagram showing the timing for the output of the parser:

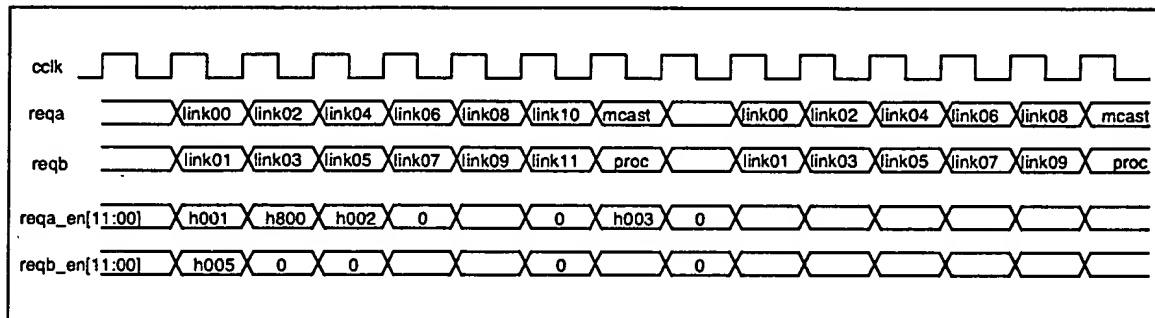


Figure 7-8: Request Parser output timing

Request Elements are valid for 1 clock cycle coming out of the parser. Along with the request elements there is a 12 bit vector which indicates which of the 12 output links the request element is destined for. In the figure above, req(a/b) bus holds the request elements, the notations indicate which input link they came from. The figure shows a strict encoding of the links, however due to design requirements of the request arbiter 'reqa' will always have the higher priority request. So, for timing on the first request element, link00 and link01 will always be written into the buffer pool first, but the parser may switch which link is output on reqa- if link01's request was a higher priority it would go out on reqa.

Request which come into the parser may be made available on the next clock cycle or as great as 7 clock cycles later. It depends on which input link they came in on, and the current input link pair that the parser is working on.

The top level signal I/O for the request parser is given below:

Signal Name	Width	I/O	Note
cclk	1	I	Core clock -250MHz
resetx	1	I	Active low module reset
eor_en	1	I	End of Row
ilink_mask[11:00]	12	I	Input Link Wiring Configuration
stagenum[11:00]	3	I	Input Link Stagenumbers
req[11:00]	52	I	Request Elements from input links #0-#11
req_start[11:0]	1	I	Request Element Enable from input links
start_align_error	1	O	Start Signals not aligned, or 1 non-existent.
start_min_error	1	O	Start Signals violated minimum request element spacing

Proprietary and Confidential Information of Onex Communications Corporation

link_dest_error,	1	0	Element attempted to violate link Wiring Configuration
dest_capture	52	0	The Element which violated the wiring configuration
req_mcast	52	1	Multicast request message.
req_uproc	52	1	Tensilica request message
mcast_link_en	12	1	1 Bit per output link that the req should goto
uproc_link_en	12	1	1 Bit per output link that the req should goto
reqa	52	0	Parsed Request Element
reqb	52	0	Parsed Request Element
reqa_en	12	0	Enable flags for the 12 output links for request a
reqb_en	12	0	Enable flags for the 12 output links for request b

7.2.1.1.1 Request Parser Design Notes

The request parser latches the logical pulse of the 'request start' signal to indicate whether or not the input link has a valid request. The rate of these start signals is equal to or greater than the processing time needed to write the requests into the request buffers. Since the request logic can write the buffers into the buffer pool 2 at a time, on every clock cycle, a 3 bit free running counter will examine the 'request valid signals' and process the request.

1. Latch the logical 'request start' pulses which occur at a maximum rate of 1 every 8 clock cycles. This signal goes high for 1 clock to indicate that a new request element has been assembled.
2. Every clock cycle look at 2 of these latched request starts. Input link 0 and 1 will be processed first, then 2 and 3 and so on. If the request start latched signal is not set, this input link will be considered 'idle' and not be processed.
3. Pull out the 4 bit self routing tag which is 'valid' for this input's stage number. Replace with the input port it came in on. Do this for both elements. There shall be several programmable registers which indicate to the parser which stage of the switch that particular input link is a member of.
4. Sort the 2 request elements priority so that the 'a' request element output is the higher of the 2 priorities. (this is a requirement of the buffer pool).
5. Register the request elements along with the signals which indicate which output port it is destined for These are 'valid signals' for the 2 request elements.
6. Clear the latched Request Start signal.
7. Increment Counter.
 - Each input stage needs to have a 3 bit field programmed which sets the stage number that the input link is located at. This allows the switch fabric to exist in multiple stages of the switch fabric.
 - Multicast and In band control messages also input a 12 bit field which indicates which of the 12 output links it should go out on.
 - At the beginning (or end of) every row CLEAR all of the latched signals so that the following row time doesn't have the previous rows request elements.

Proprietary and Confidential Information of Onex Communications Corporation

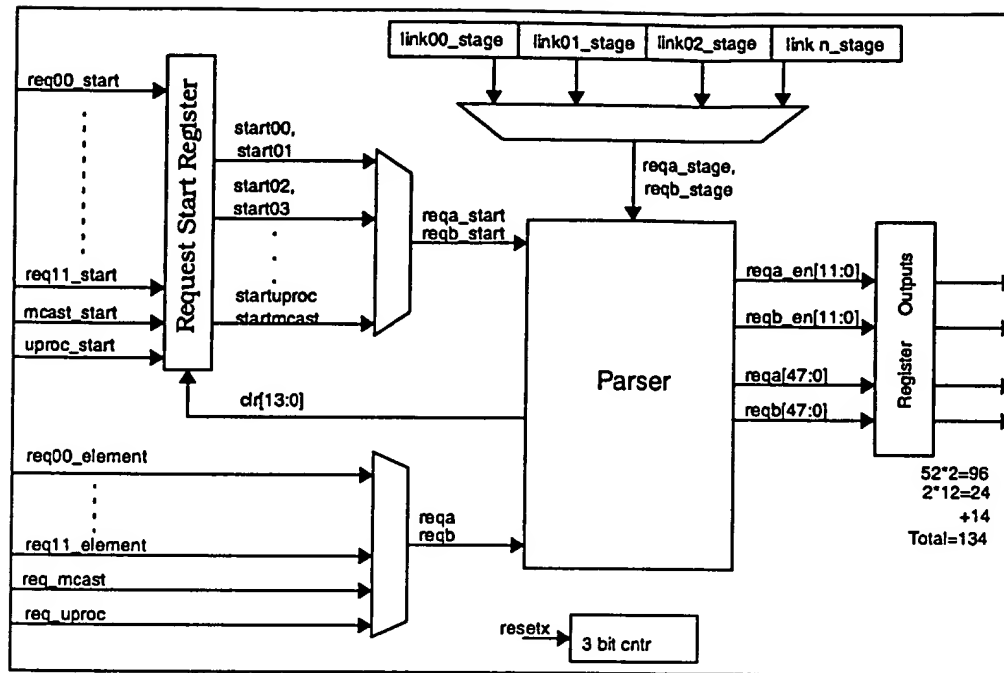


Figure 7-9: Request Parser Block Diagram

7.2.1.2 Request Arbiter

The request arbiter will be instantiated once per output link. It will connect to the request parser. The purpose of the request arbiter is to provide a small pool of requests (24) from which the highest priority oldest stored request will always be made available to the row mapper module. The input timing to the request arbiter matches that of the parser. The worst case output timing is given below:

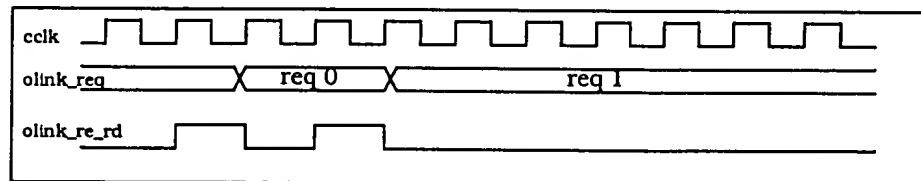


Figure 7-10: Request Arbiter Output Timing

The output logic must be able to supply a new request element every 2 clock cycles.

When a request becomes 'valid,' It is written to the buffers on the following clock cycle. After it has been written into the buffers, it is available to be output from this module on the next clock cycle.

Requests are written to the buffer pool by using the following rules (request A always has priority equal to or greater than B):

1. Request A always will have a priority equal to or greater than request B.
2. Request A will be resolved before Request B w.r.t. free buffers and evictions.
3. If there are any empty buffers, the requests are written into them before anything is evicted. Request A will be allocated to the first free buffer, if another free buffer exists request B will be granted it.
4. If there are not enough free buffers, the lowest priority youngest request which is in the buffer pool is evicted if its priority is less than that of the incoming request elements.
5. Situation: Request A is priority 5, request B is priority 3, there is 1 free buffer and the lowest

Table 7-11: Request Arbiter Top Level I/O

7.2.1.2.1 Request Arbiter Design Notes

The Buffer Pool will consist of buffering the actual request elements as well as a TimeStamp field which is unique per priority class. This allows the buffer pool manager to know which request elements can be overwritten (lowest priority, newest requests), and which request elements should be forwarded to the next switch element (highest priority, oldest requests) . Below is an illustration of the buffer:

5	58	52	51	48	47	44	43	40	39	36	35	32	31	28	27	24	19	12	11					8	7	6			2	1	0
F	TimeS- tamp		Stage 1		Stage 2		Stage 3		Stage 4		Stage 5		Stage 6		Stage 7		ReqID		Output Port ID				num	un - used	Priority						

Table 7-12: Buffer Pool Storage Element

As requests come streaming into each output link buffer pool, there will be a controller which determines if the request should be dropped, written to an unoccupied buffer or overwrite a particular buffer which contains a lower priority request. Each request will be given a timestamp unique to its priority class as well as a timestamp/priority pair which is used to search the buffer pool. This pair will be compare against the contents of each of the buffers in the buffer pool. The buffer searches are completely decoupled from one another so that this will not be a critical path which needs to ripple across all 24 buffers. In the event that a buffer is empty another set of signals will uniquely identify that buffer, and the request element will be written into it. For each request element here are the pieces of information which will be given to each request element:

- **OverWrite TimeStamp** - This is the timestamp which will be used to search the buffer pool
- **Current TimeStamp** - This is the timestamp which the request element has been assigned
- **RequestWriteEnable** - Active hi, indicates to search the buffer pool using the OverWrite Time Stamp
- **Empty Buffer Enables[23:0]** - A vector with only 1 of its bits active which assigns the request element to a particular buffer which is known to be empty.

This information will be used by the buffer pool 'write' logic which is shown below.

Proprietary and Confidential Information of Onex Communications Corporation

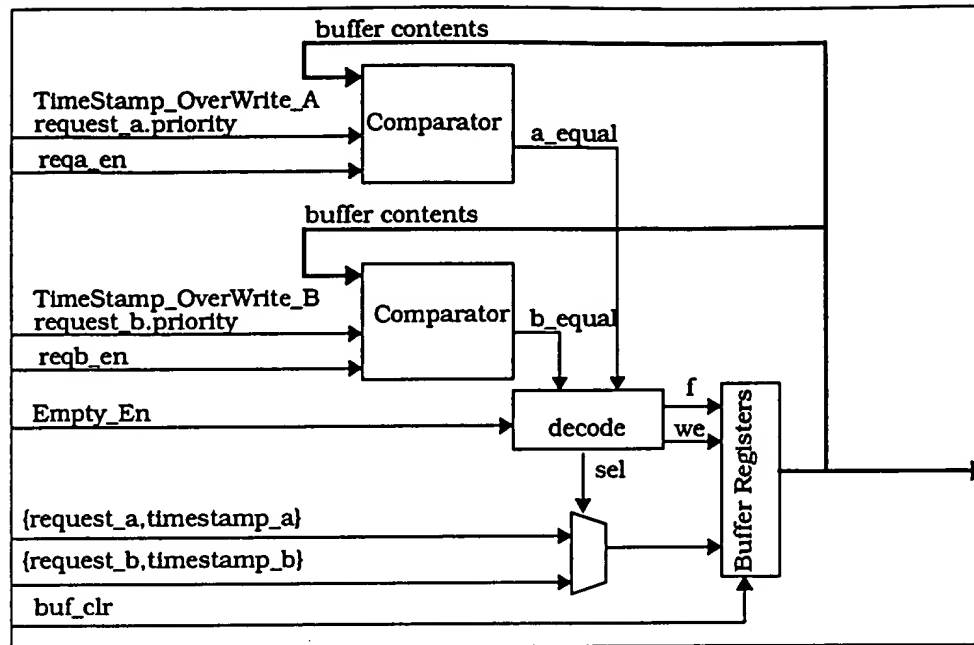


Figure 7-13: Buffer Pool 'Write' Logic

The buffer pool write logic will be controlled by a buffer pool manager. This buffer pool manager will need to keep track of the timestamps as well as which buffers are empty. The following counters will be kept:

- NextWrite TimeStamp (8 counters- 1 for each priority)
- NextRead TimeStamp (8 counters- 1 for each priority)

The updating of these counters and the assigning of timestamps will be governed by the following rules:

1. Whenever a new request comes, if a buffer is empty it is assigned to this request element. The Next-Write TimeStamp counter is assigned to this request and the counter is incremented.
2. If all the buffers are full and there is a lower priority request exists in the bufferpool (next read and next write timestamps for each priority are not-equal), the newest lowest priority request (NextWriteTimeStamp) will be used as the TimeStampOverWrite, and that counter will be decremented. The NextWriteTimeStamp for the request's priority will be assigned to the request element and incremented.
3. If there don't exist any free buffers and the lowest priority is greater than or equal to the current request element it will not be assigned to anything, its request enable signal will be inactive.

Timestamps will be for each request element which comes in. If it is a 2 group request element, only a single timestamp will be assigned.

Continuously, the following is re-evaluated:

- All of the NextWriteTimeStamps and the NextReadTimeStamps will be compared, if any are not equal the highest priority one will be chosen. The NextRead TimeStamp will be used along with the priority to search the buffer pool. All of the buffers are compared against these values, and the one which matches will be output.
- When the slot mapper logic needs a request element this one will be output (registered) and the

buffer cleared. When this happens the NextReadCounter will be incremented.

Each of these counters will have the ability to be cleared (at the start of a row) and be incremented or decremented by 1 or 2 every clock cycle. Since there are only 96 request (or grant) elements the NextRead Counter will never get larger than 96. Likewise, when a new request comes in the NextWrite counter gets incremented, but when a buffer pool request is overwritten, the counter is decremented. As a result, the worst case is all traffic of the same priority. The counter will go up to 96 and stay there since newer requests do not have preference over older requests at the same priority level. This is consistent with the oldest highest priority requests being forwarded to the next switch element in the fabric. The 'Next Read' timestamp is incremented each time a new buffer is played out of the buffer pool. This insures that the 'first-come-first-serve' principle remains in effect.

This module should be implemented using 2 counters for each priority (16 total counters @ 7 bits wide) for a total of 112 FFs.

Below is a block diagram showing the partitioning for the buffer pool logic.

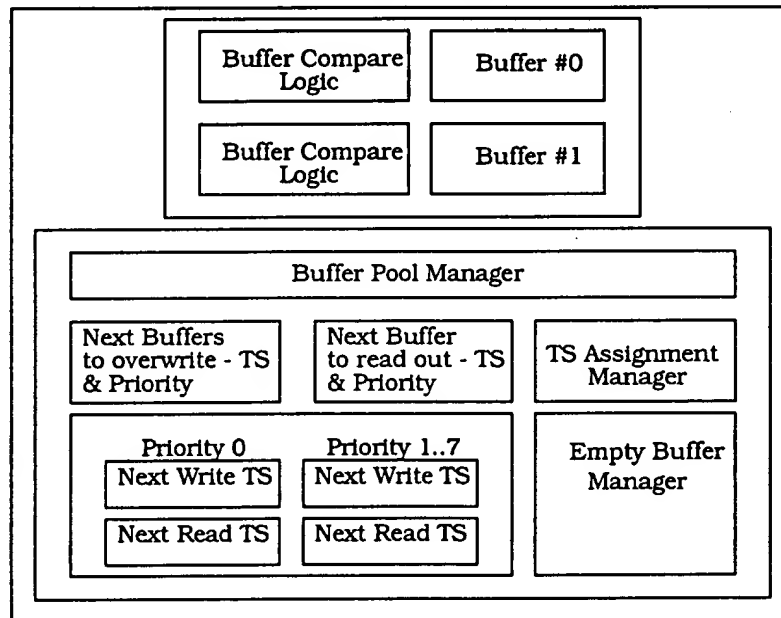


Figure 7-14: Buffer Pool Partitioning Block Diagram

7.2.2 Statistics Gathering

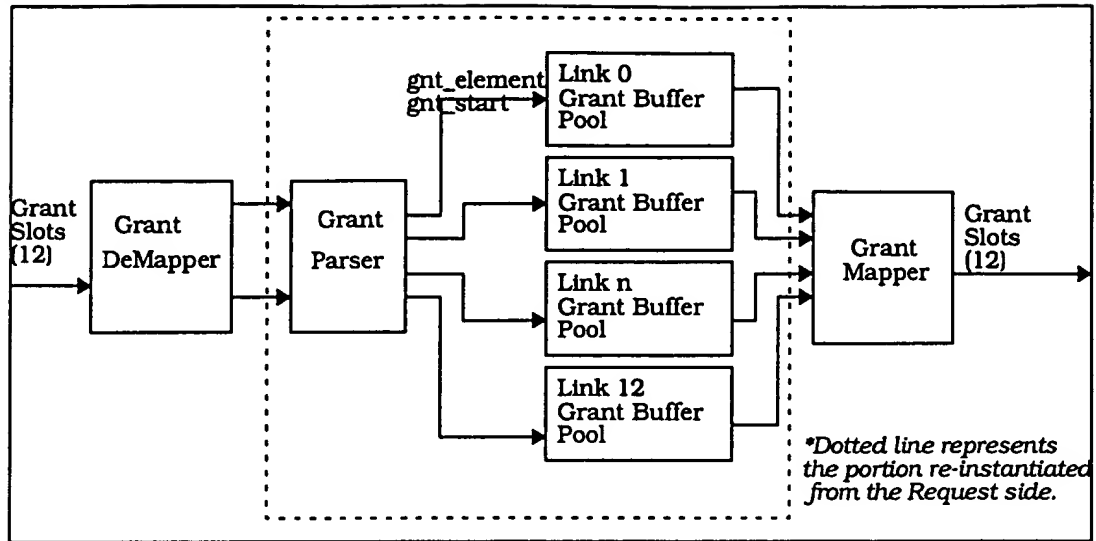
RESERVED

7.2.3 Grant Implementation

The grants flow through the switch in an out of band network. The grant processing is the same as the request processing, so the 'parser' and 'buffer pool' logic will be re-instantiated on the grant path. The only difference will be that for the grant message, some logic needs to assemble the grant elements, just like in the datapath there is a request assembler. This logic will take slot data from the synchronizers and assemble grant elements. On the output end, there will be a mapping function which takes grant elements and maps them into the correct slots. Then, these slots will be

Proprietary and Confidential Information of Onex Communications Corporation

converted into a serial stream.



The Grant Serial link will be run at 2.2gbps, the same speed as the data path's serial links. The data path has 2 links which combine to source 1700 slots at a rate of 125M slots per second. Since the grant path has half the bandwidth it sources slots at a 75Mslot per second rate.

Grant Elements arrive at a rate of 2 per every 3 slots. In every slot time on the grant link there are 4 clocks. Therefore, for every Grant, 1.5 slots will have occurred, and there will be 6 clock cycles. Since there are 12 links, 2 grant elements need to be processed every clock cycle to keep up with the input data rate. A minimal amount of buffering will be needed to hold the assembled grants to allow them to be played out at the correct rate.

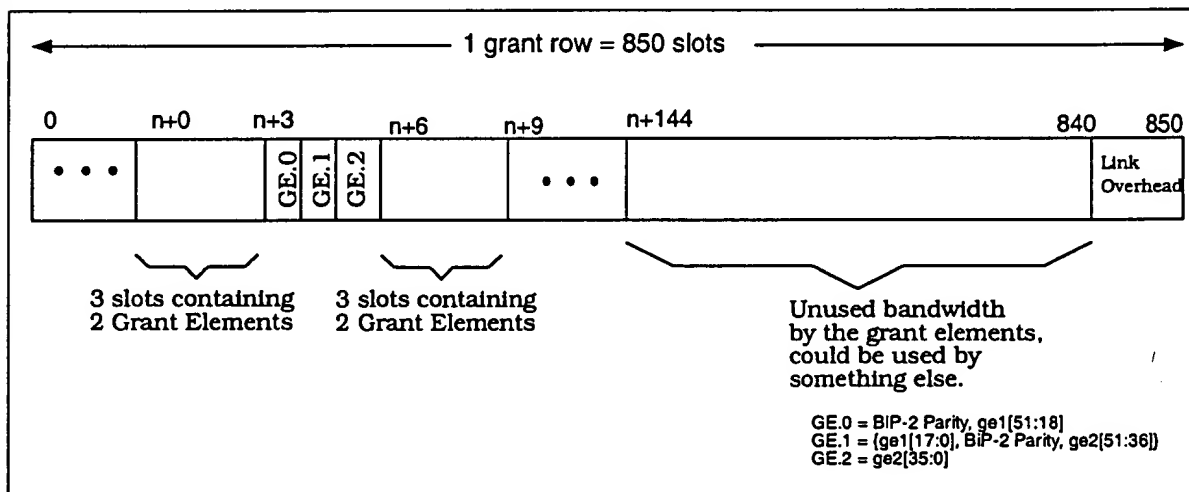


Figure 7-15: Grant Link Mapping/DeMapping Timing

The unused bandwidth between grant elements can be used to carry other traffic. Nothing is defined at the moment which uses this bandwidth (or can even have access to it), but it is nonetheless there.

A group of 3 slots comprises 2 request or grant elements. The following diagram describes the mapping as well as the parity generation of the slots to arbitration elements.

Figure 7-16: Grant/Request Element Slot Format

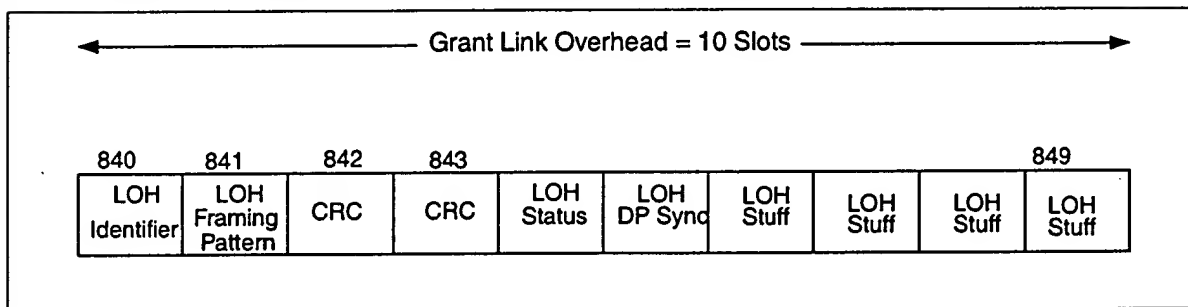
Proprietary and Confidential Information of Onex Communications Corporation

	35	34	33	17	16	00
GE.0 Slot	GE0 PO	GE0 PE	GE0 [51:18]			
GE.1 Slot	GE0 [17:00]			GE1 PO	GE1 PE	GE1[51:36]
GE.2 Slot	GE1 [35:00]					

GE_x PE= Grant Element Even Bits Parity
Calculated as $GE_x[50] \wedge GE_x[48] \dots \wedge GE_x[0]$
GE_x PO= Grant Element Odd Bits Parity
Calculated as $GE_x[51] \wedge GE_x[49] \dots \wedge GE_x[1]$

7.2.3.1 Grant Link Overhead

There will normally be 10 slots used for Link Overhead (LOH). The mapper module will be



responsible for inserting contents of most of these LOH slots into the link data stream. these 10 LOH slots into the link data stream. There are 4 types of data which may be inserted into the LOH slots:

LOH Framing Pattern -

This will be a 36-bit value which is common to all output links. It will be Configurable via a software programmable register. This pattern will be used in only 1 of the 10 LOH slots.

LOH Status -

This 32 bit status field will be configurable via a software programmable register. There will be a unique status register for each output link.

Note: the 4 tag bits are fixed to all 1's.

LOH Identifier -

This 32-bit will contain an identifier for this switch & link. The field is made up as:

- loh_id[3:0] = link number that the output mapper is instantiated as.
- loh_id[27:4] = ITSE ID number which is SW configurable.
- loh_id[31:28] = stage number the ITSE is programmed as.

Note: the 4 tag bits are fixed to all 1's.

LOH Stuff -

This 32-bit pattern will be inserted in the LOH slots which aren't used for framing, status, or ID. This pattern will be Configurable via a software programmable register and is common to all output links.

Proprietary and Confidential Information of Onex Communications Corporation

Note: the 4 tag bits are fixed to all 1's

LOH Stuff -

This is gotten from the deserializers on the datapath. The values are mapped into a 36 bit slot as follows. The actual decoding of these bits is reserved for the Synchronizers Specification. Please consult it for the use and interpretation of these bits.

loh_sync[31:26] = 0

loh_sync[25] = sync_data1_synced

loh_sync[24] = sync_data0_synced

loh_sync[23:17] = 0

loh_sync[16] = sync_offset_count_valid

loh_sync[15:0] = sync_offset_count

Note: The 4 tag bits are fixed to all 1's

CRC -

This field is inserted by the serializer unit. This is fixed, and must be the 2 slots directly following the framing pattern. The grant mapper ram should be programmed as IDLE for these slots.

7.2.3.2 Grant DeMapper

The grant DeMapper will accept slots and assemble grant elements for the grant parser. The following is the top level I/O for this module.

RESERVED

Table 7-17: Grant DeMapper Top Level I/O

The grant demapper will utilize a Ram which is addressed by the current slot number. The output of the Ram will indicate if the current slot is a valid grant element. The grant demapper ram is 32 bits wide and has the following bits:

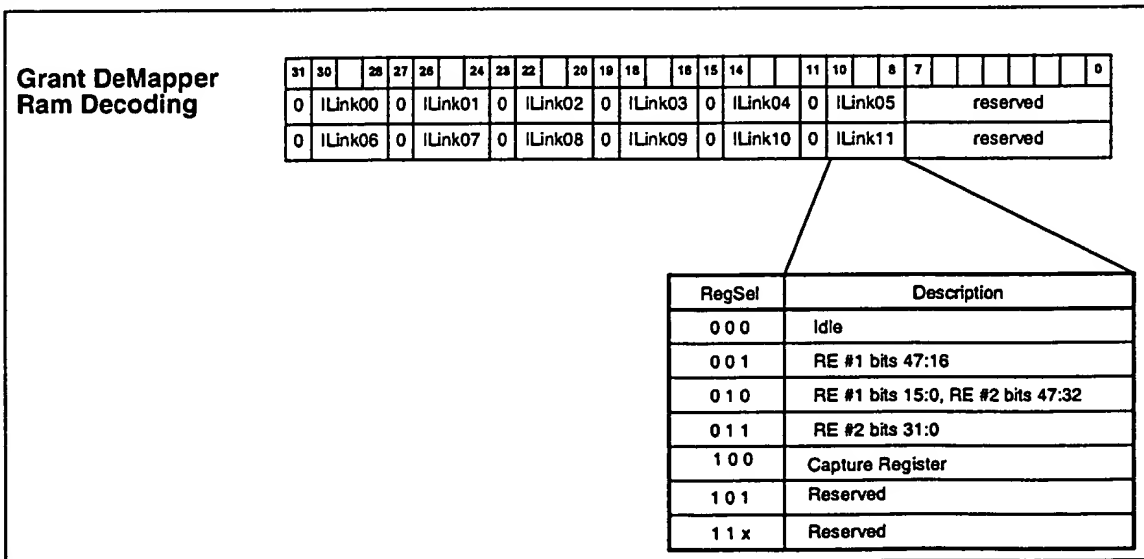


Figure 7-18: Grant DeMapper Ram

Proprietary and Confidential Information of Onex Communications Corporation

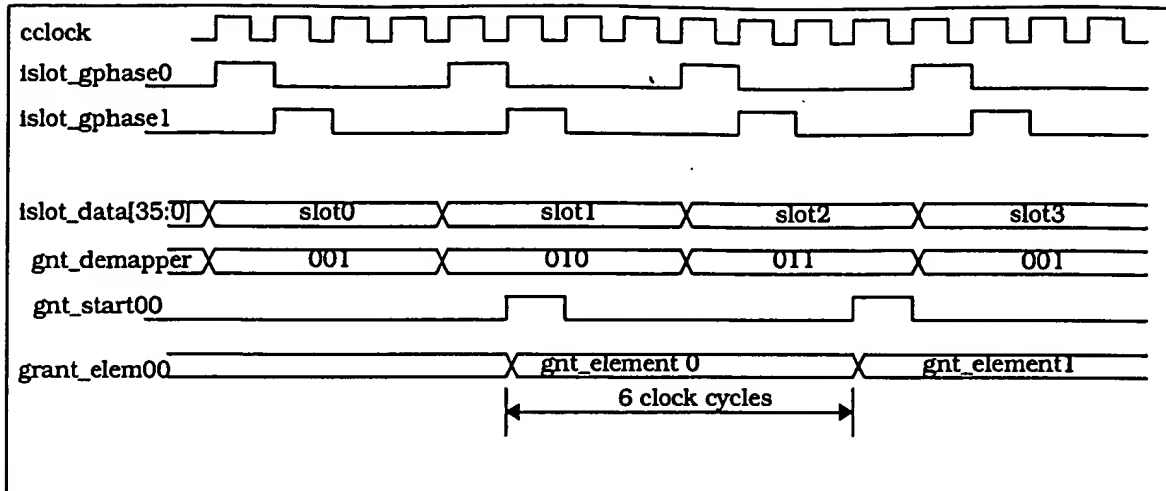


Figure 7-19: Grant DeMapper Timing

The grant demapper ram allows a switch chip to be located in more than 1 stage of a switch fabric by programming the DeMapper Ram accordingly. The ram slot counter will be the data path slot counter with the LSB dropped to account for the 50% decrease in the number of slots per row. The ram will be programmed so that different input links can have their requests valid at completely different times as long as the time between 2 incoming slots allows both to be processed by the grant logic (16 clock cycles). The demapper ram has 3 bits dedicated per input grant link so that future expansion is possible.

7.2.3.3 Mapper Ram

Here is the top level I/O for the Grant Mapper Module:

RESERVED

Table 7-20: Grant Mapper I/O

Table 7-21:

Proprietary and Confidential Information of Onex Communications Corporation

Grant Mapper
Ram

31			28	27			24	23			20	19			16	15			11	10			8	7					0		
ILink00				ILink01				ILink02				ILink03				ILink04				ILink05				reserved							
ILink06				ILink07				ILink08				ILink09				ILink10				ILink11				reserved							

RegSel	Description
0000	GE IDLE
0001	GE #1 bits 47:16
0010	RE #1 bits 15:0, GE #2 bits 47:32
0011	GE #2 bits 31:0
1000	Framing Pattern
1001	Link Overhead Status Slot
1010	Link Overhead ID Slot
1011	Link Overhead 'Stuff' Slot
1100	Link Overhead Sync Slot
others	Reserved

Figure 7-22: Grant Mapper Ram

When the Mapper RAM indicates it is time to transmit a grant element, the highest priority 52-bit request element is fetched from the arbitration module. Since only 32-bits of the GE may be transmitted in a single link slot, it will be necessary to add buffering within the Output Grant Mapper module which will buffer the extra bits which cannot be sent in the current slot.

A 3-slot structure will be defined which will be used to transmit 2 52-bit GEs. The timing for fetching GEs is shown below. In this example, the 3 slot GE structure is programmed to be transmitted on output link slots N, N+1, and N+2.

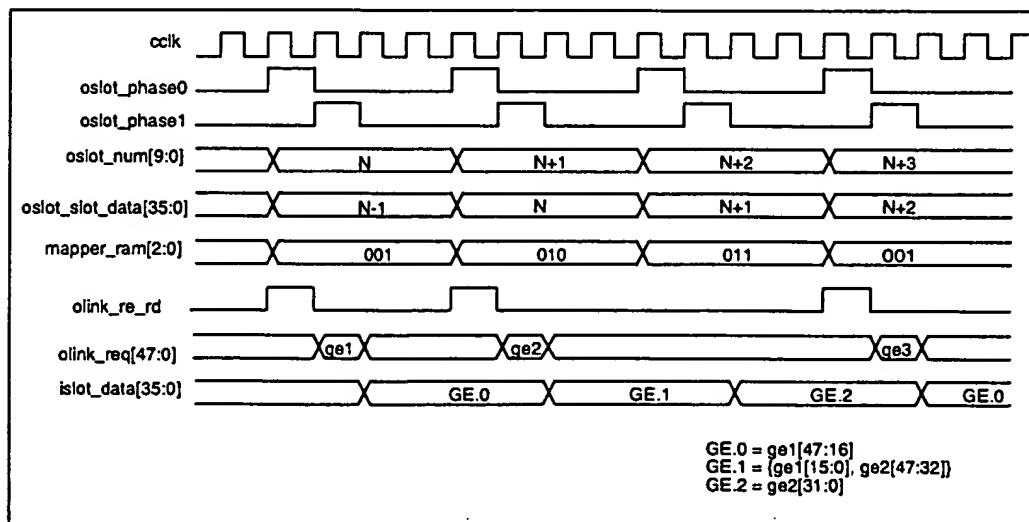


Figure 7-23: Grant Element Timing

There will be a programmable field which indicates the total number of requests or grants which can be made every row time. This value will be per output link and take into account the number of groups locked out for TDM traffic. After this value is reached, the output link will output

idles. This is independent of the configuration ram mapping. By doing this, software has an easy way to do performance testing and network congestion by setting this value to something small. It also allows a system to overrequest or over-grant a link since arbiters further on down the line will 'knock-out' a certain percentage of requests.

7.2.4 Request Arbitration Memory Map

Request Link 0 Priority 0 Statistics																		0x9000
Request Link 0 Priority 1 Statistics																		0x9004
Request Link 0 Priority 2 Statistics																		0x9008
Request Link 0 Priority 3 Statistics																		0x900C
Request Link 0 Priority 4 Statistics																		0x9010
Request Link 0 Priority 5 Statistics																		0x9014
Request Link 0 Priority 6 Statistics																		0x9018
Request Link 0 Priority 7 Statistics																		0x901C
Request Link 1 Priority 0 Statistics																		0x9020
Request Link 2 Priority 0 Statistics																		0x9040
Request Link 3 Priority 0 Statistics																		0x9060
Request Link 4 Priority 0 Statistics																		0x9080
Request Link 5 Priority 0 Statistics																		0x90A0
Request Link 6 Priority 0 Statistics																		0x90C0
Request Link 7 Priority 0 Statistics																		0x90E0
Request Link 8 Priority 0 Statistics																		0x9100
Request Link 9 Priority 0 Statistics																		0x9120
Request Link 10 Priority 0 Statistics																		0x9140
Request Link 11 Priority 0 Statistics																		0x9160
0	Max Request Link 00	0	Num Requests Link 00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x9180
0	Max Request Link 01	0	Num Requests Link 01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x9184
0	Max Request Link 02	0	Num Requests Link 02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x9188
0	Max Request Link 03	0	Num Requests Link 03	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x918C
0	Max Request Link 04	0	Num Requests Link 04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x9190
0	Max Request Link 05	0	Num Requests Link 05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x9194
0	Max Request Link 06	0	Num Requests Link 06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x9198
0	Max Request Link 07	0	Num Requests Link 07	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x919C
0	Max Request Link 08	0	Num Requests Link 08	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x91A0
0	Max Request Link 09	0	Num Requests Link 09	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x91A4
0	Max Request Link 10	0	Num Requests Link 10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x91A8
0	Max Request Link 11	0	Num Requests Link 11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x91AC

Proprietary and Confidential Information of Onex Communications Corporation

Base Address 0x00440000

31	24	23	16	15	8	7	0	Address Offset								
unmapped address space								0x5A90								
Grant Link 00-11 Status Reg								0x8000 0x802C								
Grant Link Framing Pattern								0x8030								
	Grant Link Framing Pattern		read only- all zeroes					0x8034								
Grant Link Common 'Stuff' Reg								0x8038								
Grant Link 00 Line Overhead Status Reg								0x803C								
Grant Link 11 Line Overhead Status Reg								0x8068								
Grant Max Grants								0x806C 0x8070 0x8074								
Grant Capture Interrupt Status Register								0x8078								
Grant Link 0-11 Capture Register Contents								0x807C 0x80A8								
Grant Link 0 - 11 Capture Register Masks								0x80AC 0x80D8								
Gnt Config		0	0	0	0	0	0	0	0	0	0	0	0	Grant Parity Error Masks		0x80DC
Grant Error Interrupt Mask								0x80E0								
Grant Error Interrupt Status Register																0x80E4
														Grant Parity Error		0x80E8
				Grant Mapper Sequence Error								Grant DeMapper Sequence Error				0x80EC
Grant Dest Error Element[31:00]								0x80F0								
Grant Dest Error Element[47:32]]				reserved				0x80F4								

Table 7-24: Grant Memory Map (Shamelessly copied from Ch. 15)

7.2.5.1 Grant Mapper RAM

31	28	27	24	23	20	19	16	15	12	11	8	7	0	Address Offset
Grant Link 00	Grant Link 01	Grant Link 02	Grant Link 03	Grant Link 04	Grant Link 05	0	0	0	0	0	0	0	0	0x0000
Grant Link 06	Grant Link 07	Grant Link 08	Grant Link 09	Grant Link 10	Grant Link 11	0	0	0	0	0	0	0	0	0x0004
Unknown														Reset Value

The Grant Mapper Ram specifies the outgoing slot numbers that grant elements will be put upon. It is critical that this ram be written before the switch fabric is made operational. The reset value of this register is unknown. Any location may be written to or read at any time by the master

Proprietary and Confidential Information of Onex Communications Corporation

processor or the host interface. However, it is recommended that the processor update the values of this ram when traffic is either not going through the link, or during a slot location far away from the current slot number. Bits 7-0 will always read back a zero.

Mapper Ram Bit Coding	
Value (0b)	
0000	Idle
0001	GE.0
0010	GE.2
0011	GE.3
1000	LOH Framing Pattern
1001	LOH Status
1010	LOH ID
1011	LOH Stuff
1100	LOH Sync

7.2.5.2 Grant DeMapper RAM

31	28	27	24	23	20	19	16	15	12	11	8	7	0	Address Offset
0	Grant Link 00	0	Grant Link 01	0	Grant Link 02	0	Grant Link 03	0	Grant Link 04	0	Grant Link 05	0	0	0x4000
0	Grant Link 06	0	Grant Link 07	0	Grant Link 08	0	Grant Link 09	0	Grant Link 10	0	Grant Link 11	0	0	0x4004
Unknown														Reset Value

The Grant DeMapper Ram specifies the incoming slot numbers that grant elements will be coming in upon. It is critical that this ram be written before the switch fabric is made operational. The reset value of this register is unknown. Any location may be written to or read at any time by the master processor or the host interface. However, it is recommended that the processor update the values of this ram when traffic is either not going through the link, or during a slot location far away from the current slot number. Bits 7-0 will always read back a zero.

Mapper Ram Bit Coding	
Value (0b)	
000	Idle
001	GE.0
010	GE.2
011	GE.3
1000	Capture Register

Proprietary and Confidential Information of Onex Communications Corporation

7.2.5.3 Grant Link Status Register

31	29	24	23	22	16	15	14	8	7	6	0	Address Offset																	
fifo fill	0	Current Fifo Watermark						0	Grants Received Last Row						0	Grants Dropped Last Row						0	Grants Forwarded Last Row						0x8000-0x802C
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Reset Value

There is a grant link status register for each of the 12 outgoing links. Address offset 8000 is for link 0, while 802C is for link 11. The link status register is read only. Writes do not have an effect on these registers. The sub fields are defined as:

FifoFill: This bit shall be set if the grant fifo is currently filled.

Current Fifo Watermark: This shall read back the current number of grants that are buffered.

Grants Received Last Row: Total number of grants received during the last row. This is updated every row upon an EndOfRow. This is independent of the number of PDUs which the grant was supposed to reserve.

Grants Dropped Last Row: Total number of grants dropped during the last row. This is updated every row upon an EndOfRow. This is independent of the number of PDUs which the grant was supposed to reserve.

Grants Forwarded Last Row: Total number of grants which were forwarded during the last row. For GE's which reserve more than 1 PDU this is the total number of PDUs reserved.

7.2.5.4 Grant Link Framing Pattern

31	27	24	23	16	15	8	7	0	Address Offset																					
Frame0									0x8030																					
0	0	0	0	Frame1				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x8034

The grant link framing pattern is 36 bits wide and is made by concatenating frame0 with fram1 such that: FramingPattern[35:0] = Frame1[27:24],Frame0[31:0]. The reset value of Frame 0 is 0x0F628_0000 and Frame 1 is 0. So that the framing pattern is 0x0F628_0000.

7.2.5.5 Grant Link Common 'Stuff' Reg

31	27	24	23	16	15	8	7	0	Address Offset
Stuff Reg									0x8038
0	0	0	0	0	0	0	0	0	Reset Value

This register's contents will be inserted into the link overhead 'stuff' slot designated by the mapper ram. This register is read/writable at any time. Writes are effective immediatly.

Proprietary and Confidential Information of Onex Communications Corporation

7.2.5.6 Grant Link Line Overhead Status Register

31	27	24	23	16	15	8	7	0	Address Offset
Link 0-11 Overhead Status Register									0x803C 0x8068
0	0	0	0	0	0	0	0	0	Reset Value

This register's contents will be inserted into the link overhead's status slot designated by the mapper ram. This register is read/writable at any time. Writes are effective immediatly.

7.2.5.7 Maximum Grants Register

31	27							24							23	16							15	8							7	0							Address Offset
Link 00 Maximum Grants								Link 01 Maximum Grants								Link 02 Maximum Grants								Link 03 Maximum Grants								0x806C							
Link 04 Maximum Grants								Link 05 Maximum Grants								Link 06 Maximum Grants								Link 07 Maximum Grants								0x8070							
Link 08 Maximum Grants								Link 09 Maximum Grants								Link 10 Maximum Grants								Link 11 Maximum Grants								0x8074							
0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	Reset Value						

The Maximum Grants Register may be written/read to at any time. It's reset value is 96 for each link (corresponding to the maximum number of grants which any of the links can accomodate). It is recommended that the number of groups available for PDU traffic at any given time be written into these positions to ensure that no mapper/demapper programming error or service processor error allow the introduction of more grants than PDUs that can be carried on the data link.

7.2.5.8 Grant Capture Interrupt Status Register

31	27	24	23	16	15	8	7	0	Address Offset
0	0	0	0	0	0	0	0	0	Grant Capture Interrupt Status Register
0	0	0	0	0	0	0	0	0	Reset Value

If any of the grant capture registers change from one row time to the next, the corresponding link bit in this register will get set. Bit 0 is for link 0, bit 11 for link 11. The other bits are unused and should always read back zero. Once the bit has been set, software must write a logical 1 to it to clear the offending bit. Whenever any of these bits are set the capture interrupt will become active. To clear the interrupt, the processor must clear the offending bit in this register by writing a 1 to it. Writes of 'zero' have no effect.

7.2.5.8.1 Grant Link Capture Registers

31	27	24	23	16	15	8	7	0	Address Offset
Grank Link 0 - 11 Capture Registers									0x807C 0x80D8
0	0	0	0	0	0	0	0	0	Reset Value

These are 12 read only registers which have reset values of zero. Each time the grant demapper encounters a 'Link Overhead Capture' Entry the slot data is stored in this register. These registers may be written to at any time, writes have no effect.

Proprietary and Confidential Information of Onex Communications Corporation

7.2.5.9 Grant Link Capture Mask Registers

31	27	24	23	16	15	8	7	0	Address Offset
Grant Link 0 - 11 Capture Registers									0x80AC 0x80D8
0	0	0	0	0	0	0	0	0	Reset Value

These 12 registers provide bit masks for the capture interrupt register. If any bit is set to a 1, then the hardware will monitor that bit in the link's capture register from row to row. If that bit changes, the grant link capture change register will latch the link whose bit has changed and output an interrupt to the processor. Every bit in each of the 12 input links can be set or cleared individually to trigger the capture interrupt. This register's reset value is all zero's which inhibit the interrupt from occurring. The register may be written/read to at any time.

7.2.5.10 Grant Configuration / Grant Parity Error Mask Register

31	27	24	23	16	15	8	7	0	Address Offset
Grant Configuration Register									0x80DC
0	0	0	0	0	0	0	0	0	Reset Value

Grant Parity Error Masks - bit is active hi to enable parity conformance of incoming grants.

Bit 0: Link 00

Bit 1: Link 01

Bit 2: Link 02

...

Bit 11: Link 11

Grant Config

Bit 7: Grant Rotate Enable: Reset Value is 0, Set to a 1 to enable grant parser rotation.

Bit 6: Disable Num Field (grants are forced to single PDU reservation mode). Default enables grants to reserve more than 1 PDU.

7.2.5.11 Grant Error Interrupt Mask Register

31	24	23	12	11	0	Address Offset
Grant Error Interrupt Mask						0x80E0
0	0	0	0	0	0	Reset Value

Bit 31: Grant Start Signals Unaligned Error Mask: Program to 1 to enable an interrupt to occur if link demappers are out of sync with respect to the start signals which they create. this type of error.

Bit 30: Grant Minimum Start Pulse Error Mask: Reset is 0, Program to 1 to enable this type of error. This error mask is ineffective for grants since there is no minimum pulse period (it is a holdover from the request parser).

Bit 29: Grant Remaining Mask: At the end of a row, if grants are remaining in the buffers, this will trigger an interrupt when set to a 1, reset value is 0.

Bit 28: Grant Fifo Filled Mask: If the grant fifo overflows for a link this will trigger when programmed to a 1, reset value is 0. Fifo watermarks need to be investigated for the link.

Bits 23-12: Grant Mapper Sequencing Mask: Set to a 1 to allow sequencing errors to generate an interrupt. Bit 23 refers to Link #11, bit 12 refers to Link #0.

Bits 11-0: Grant Demapper Sequencing Mask: Set to a 1 to allow sequencing errors to generate an interrupt. Bit 11 refers to Link #11, Bit 0 to Link #0.

7.2.5.12 Grant Error Interrupt Status Register

31	24	23	12	11	7	0	Address Offset
Grant Error Interrupt Status Register						reserved	0x80E4

Bit 31: Grant Start Align Error - signals alignment error, write a 1 to clear this interrupt source

Bit 30: Grant Start Min Error - signals that grant elements have arrived too quickly. Write a 1 to clear this interrupt source.

Bit 29: Grants Remaining - signals grants are still in the fifo at the end of the row. Write a 1 to clear this interrupt source.

Bit 28: Grant Fifo Filled - signals that a grant fifo has overfilled. Write a 1 to clear this interrupt source.

Bit 27: Grant Destination Error - signals that a grant element has attempted to goto an output it isn't supposed to. Write a 1 to clear this type of interrupt source.

Bit 26: Grant Parity Error - signals to read the grant parity error register

Bit 25: Grant Mapper Sequencing Error - signals to read grant sequencing error register

Bit 24: Grant Demapper Sequencing Error - signals to read grant sequencing error register

7.2.5.13 Grant Parity Error Register

31	24	23	12	11	7	0	Address Offset
reserved						Grant Link Parity Status	0x80E8

Bit 11: Link 11

Bit 10: Link 10

....

Bit 0: Link 0

When a grant parity error is detected, the register should be read to determine which link had the error. Writing a 1 to the bit in this register which is causing the interrupt, will clear the interrupt as well as the bit.

7.2.5.14 Grant Sequencing Error Register

31	24	23	16	12	11	7	0	Address Offset								
0	0	0	0	Mapper Sequencing Status				0	0	0	0	DeMapper Sequencing Status				0x80EC

When a Sequencing Error is detected, this register should be read to determine the input or output link which has generated the error. Write a 1 to the offending bit to clear the interrupt.

Proprietary and Confidential Information of Onex Communications Corporation

Bit 24: Output Link 11

....

Bit 16: Output Link 0

Bit 11: Input Link 11

....

Bit 0: Input Link 0

7.2.5.15 Grant Destination Error Register

31	24	23	16	15	8	7	0	Address Offset
Stored Grant[47:16]								0x80F0
Stored Grant[15:0]				reserved				0x80F4

This read only register allows the system to read back a grant which was caused by a wiring configuration error. Any grant which does not adhere to its input link 'imask' configuration will be latched in this register and cause an interrupt (if enabled to do so). If multiple links have errors on them, only the 'last link' (ie higher numbered) will be captured.

7.2.6 Software Notes

RESERVED

Proprietary and Confidential Information of Onex Communications Corporation

8 Control Protocol

RESERVED.

8.1 Host Interface

RESERVED.

8.2 In-Band

RESERVED.

8.2.1 Message Format

RESERVED.

8.2.2 Method of Operation - Transmitting ASIC

RESERVED.

8.2.3 Method of Operation - Receiving ASIC

RESERVED.

9 Clock Definitions

The Switch ASIC has four main functional clock domains: `core_clk`, `global_byte_clk`, `link_byte_clk`, and `link_serial_clk` (The Switch also supports JTAG and the associated `TCLK` domain). These clocks are generated from two primary input reference clocks: `link_ref_clk` and `sw_ref_clk`. A top level view of the clock domains is shown in Figure 9-1. The `link_serial_clk` clocks are generated within the receive core and transmit core. This are shown in Figure 9-2 and Figure 9-3.

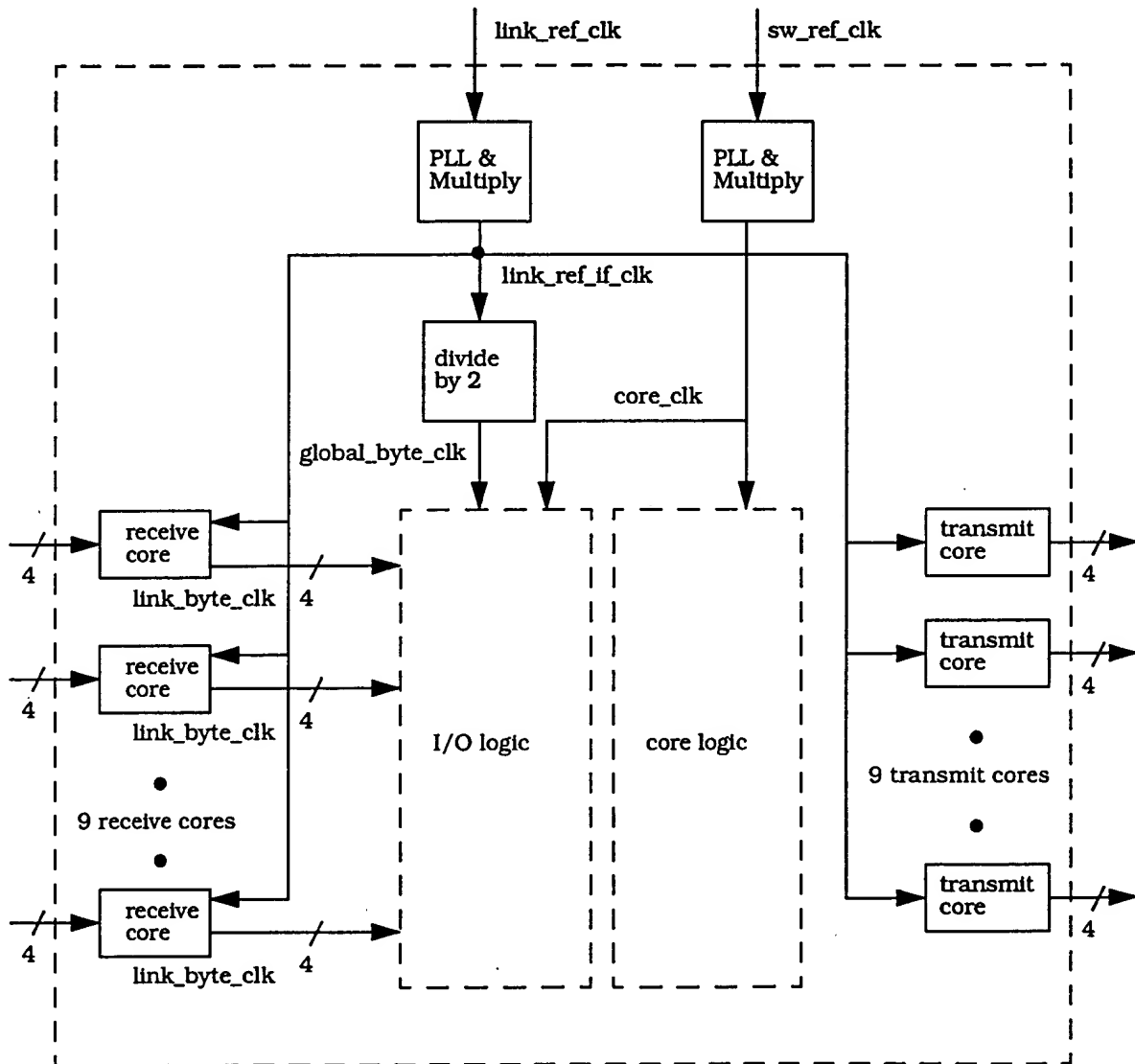


Figure 9-1: Switch ASIC Clock Domains

Proprietary and Confidential Information of Onex Communications Corporation

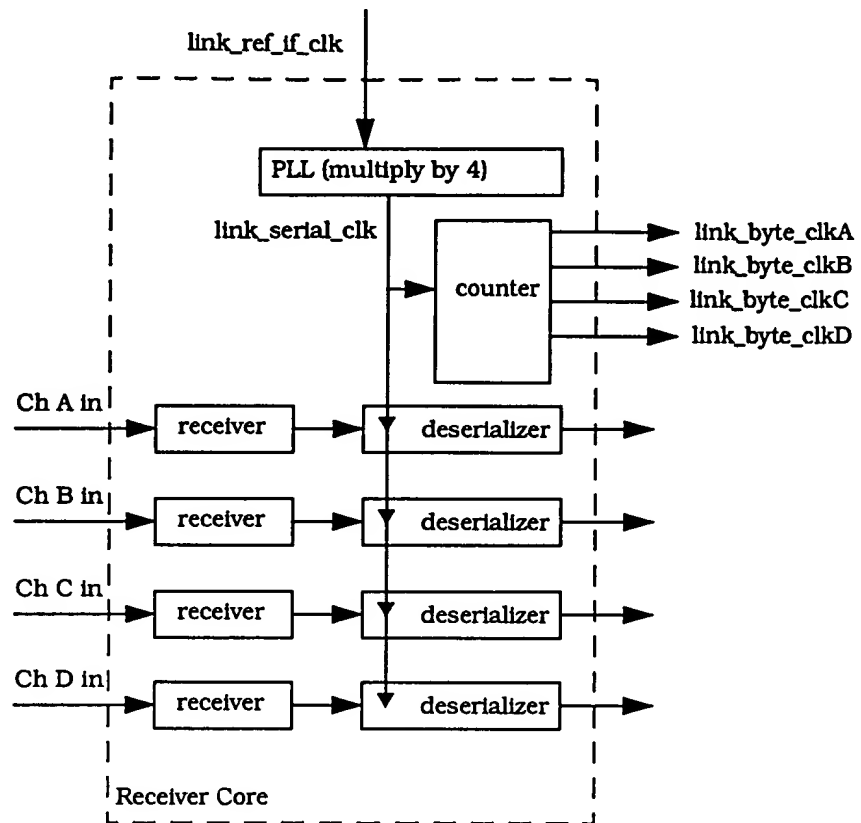


Figure 9-2: Receiver Core Clock Scheme

Proprietary and Confidential Information of Onex Communications Corporation

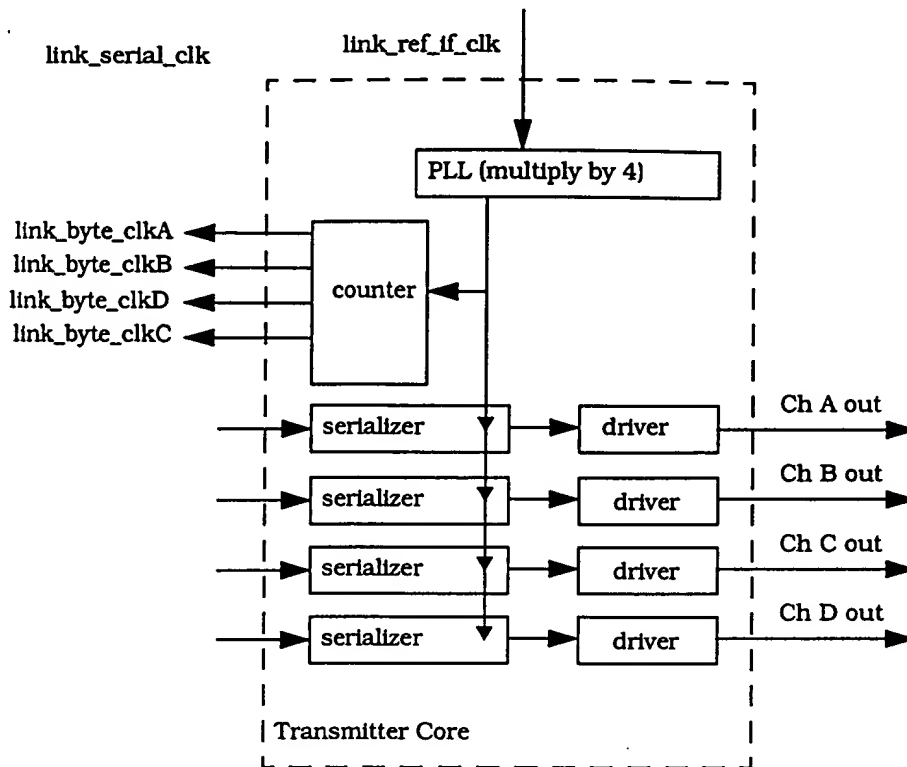


Figure 9-3: Transmitter Core Clock Scheme

9.1 Sw_ref_clk

This is the reference clock used to generate core_clk. This clock is multiplied by TBD to create core_clk.

9.2 Link_ref_clk

This is the reference clock used to generate link_ref_if_clk. This clock is multiplied by TBD to create link_ref_if_clk. The selection of this clock is based on the desired frequency for the link_serial_clk. There are, however, two additional requirements. The link_ref_clk must have a frequency that is an integer multiple of 72 KHz and must be frequency locked to the generation of the global synchronization signal sor_sync. The need for this requirement is discussed in Section 10.

9.3 Link_ref_if_clk

This clock is used as a reference clock by the transmitter and receiver cores to generate the serial clocks and the byte clocks. As is shown in Figure 9-2 and Figure 9-3, the transmitter and receiver cores contain a multiply by 4 of the incoming reference clock. Therefore, link_ref_if_clk must be operating at 1/4 the rate of the desired serial interface frequency. Assuming the following:

- 72 KHz row time
- 1700 slots (@36 bits per slot) per row
- row data is parsed across 2 serial lines

With the above assumptions, the minimum serial frequency is 2.2032 GHz. This implies a link_ref_if_clk frequency of $2.2032/4 = 550.8$ MHz.

9.4 Link_serial_clock

The link_serial_clock is generated within each transmitter and receiver core. As discussed

Proprietary and Confidential Information of Onex Communications Corporation

above, this clock must have a minimum frequency of 2.2032 GHz to support the data throughput requirements.

9.5 Link_byte_clk

The link_byte_clk is generated within each transmitter and receiver core. Each serial input or output channel generates a distinct link_byte_clk. This clock operates at 1/8 of the link_serial_clock and is used to latch the byte data to the transmitter core and from the receiver core. All link_byte_clk's are frequency locked, but there is no guaranteed phase relationship.

9.6 Global_byte_clk

The global_byte_clk is generated by dividing link_ref_if_clk by 2. Thus, it is the same frequency as the link_byte_clk's, but there is no phase relationship. However, because they are sourced from the same reference, global_byte_clk and all the link_byte_clk's are frequency locked.

9.7 Core_clk

RESERVED

Proprietary and Confidential Information of Onex Communications Corporation

10 Synchronization

Within the iTAP architecture, there are two levels of synchronization. The lower level is a point to point synchronization between serial channels. Each serial channel transmits a framing pattern at a user programmable rate (the default is 72 MHz). Each receiving serial channel will then synchronize itself to the framing pattern. Once this occurs, the serial channel is considered synchronized. More specific details of this level of synchronization are provided in Section 11.2. A higher level of synchronization involves synchronizing multiple data links within the Switch itself, and at the highest level synchronizing all of the Switches and Port Processors. The methodology to do that is discussed in this section.

The iTAP architecture requires that row data move across the switch fabric in lock step fashion. During each row time (72 MHz), a Switch ASIC will be transmitting the previous row's data and receiving the next row's data. All switching within the Switch ASIC is based on the fact that the input row data is slot aligned prior to switching. To achieve global synchronization, a globally distributed synchronization pulse, *sor_sync*, is provided to all Switch ASICs. (The Port Processor ASICs are synchronized with the assistance of the Switch ASICs, see Section 10.5). *Sor_sync* is a 72 KHz signal with the following system requirements:

- Worst case delta between any Switch ASIC receiving the *sor_sync* pulse must not exceed 16 ns.
- The minimum high time and minimum low time of the *sor_sync* pulse must be greater than two *global_byte_clk* clock cycles.
- The generation of the *sor_sync* pulse must be sourced from the same reference clock that is used to generate the *link_ref_clk* to ensure a signal that is frequency locked to *global_byte_clk*.

As mentioned above, the *sor_sync* pulse is allowed some delta in its arrival time to each Switch ASIC. With this signal being used as a reference to start the transmission of row data, it is obvious that row data arriving at a destination switch will not be slot aligned. In addition, there are possible electrical deltas in the routing of the serial lines. Because the electrical delay delta between Switch input ports effects the overall delay to compensate for, the following system requirement is placed on the serial LVDS lines to/from the Switches:

- Between any two Switches, the worst case electrical delay from a transmitting device to a receiving device must not exceed 16 ns.

In addition, the global *sor_sync* signal must be synchronized to each Switch ASIC. This may result in an additional one *global_byte_clk* cycle of difference between Switch ASICs. All of these possible worst case deltas are summed up and illustrated in Figure 10-1. A pictorial interconnect is shown in Figure 10-2.

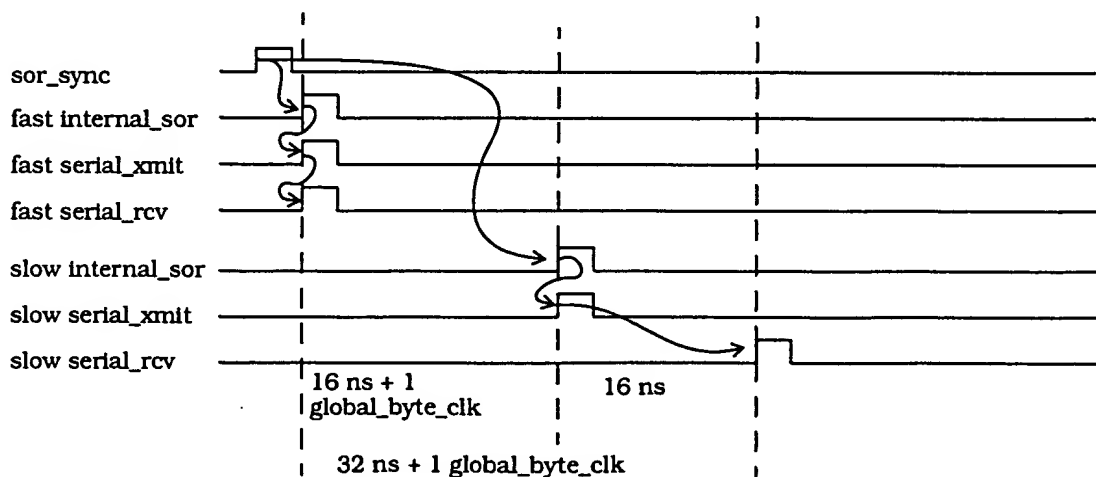


Figure 10-1: Worst Case Serial Data Arrival at Switch

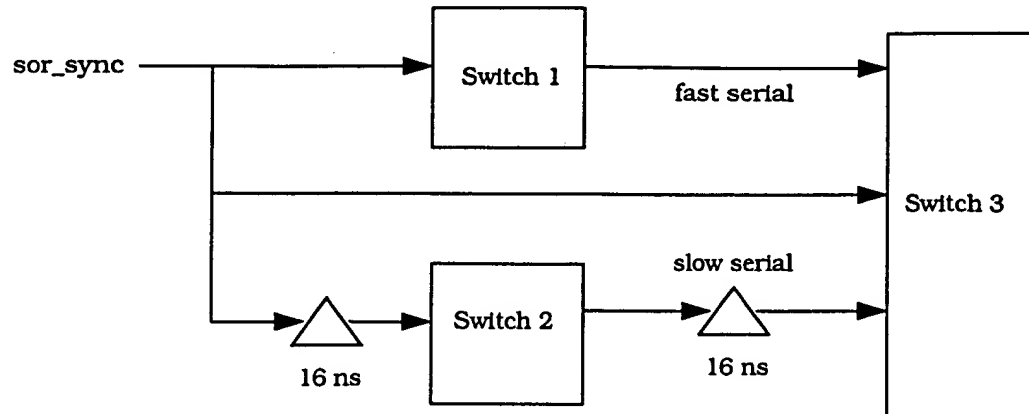


Figure 10-2: Worst Case Switch Layout Scenario

To compensate for the delta in the receipt of serial data, the Switch ASIC uses buffering logic to store data from each input port until it is guaranteed that all input ports have started to receive data. At this point in time, data can begin to be switched. This, however, creates an additional problem. By delaying the start of the switching relative to the start of row implies that an entire row of data cannot be switched by the end of the row time. The solution is that the link overhead field of the row is not switched. The link overhead field contains information that is used between connecting devices only and is generated by the output port logic.

10.1 Internal Synchronization

Once the Switch ASIC is operational (i.e. following PLL spin up, Tensilica boot, etc.), the input port control logic will be instructed from the core to initiate its synchronization. To do this, the Switch ASIC samples the `sor_sync` signal in the `global_byte_clk` domain. When the first rising edge is detected on the `sor_sync` signal, the Switch ASIC will start an internal counter, `switch_sor_counter`. Each row time, this counter counts from 1 to `ROW_SIZE` where `ROW_SIZE` is the number of bytes transferred on a serial channel during one row time. Several internal events are triggered based on the value of this counter. A summary of the programmable registers used to control these events can be found in Table 11-11.

Due to electrical changes at the system level (based on voltage, temperature, etc.) and the fact that the `sor_sync` signal is asynchronous to `global_byte_clk`, it is possible that the relative position of the rising edge of the registered version of `sor_sync` compared to the rising edge of `switch_sor` may vary from row to row. However, because the `sor_sync` signal is frequency locked to `global_byte_clk`, the delta should never exceed ± 2 `global_byte_clk` cycles. Error logic within the port control logic will monitor the delta and set an error flag if the delta exceeds a user programmable value.

Once the Switch ASICs have started their respective `switch_sor_counters`, they are all synchronized to each other within a predictable worst case delta. This worst case delta is equal to 16 ns plus one `global_byte_clk` period. The 16 ns is from the maximum delta on the `sor_sync` line (see above). The `global_byte_clk` period is due to the fact that the Switch ASIC port clocks are asynchronous to each other and asynchronous to the `sor_sync` signal.

10.2 Data Channel Synchronization

To achieve the necessary bandwidth, the data link is split across two serial lines or channels. The first stage of synchronization within the Switch is to synchronize each pair of channels. This compensates for any electrical delta between the channels that make up a link. This also places a

Proprietary and Confidential Information of Onex Communications Corporation

requirement on the electrical delta between the serial channels that make up a link:

- Between any two serial data channels that make up a data link, the worst case electrical delta must not exceed 2 ns.

The data alignment of two serial channels is accomplished using a ring buffer and is discussed further in Section 11.2.

10.3 Slot Data Synchronization

As mentioned previously, within the Switch ASIC all input ports must present synchronized slot data to the core. Each device (Switch and Port Processor) will transmit serial data based on its own internal sor signal. To compensate for the deltas in arrival time of incoming data as shown in Figure 10-1, each serial input channel has a data slot FIFO (see Section 11.2.2.1). As shown in Figure 10-1, the worst case delta between any input port is 32 ns plus 1 global_byte_clk. Thus, the slot data FIFO must be large enough to buffer the quantity of data that can be written in this time period.

To slot align the data inputs, all slot data FIFOs must be read simultaneously. To achieve this goal, a delayed version of the switch_sor signal is used to start the read. Based on the frequency of global_byte_clk and the information presented above, it is possible to determine a delay value relative to the switch_sor signal when slot data will be available from all input ports. For example, Figure 10-3 shows that for the worst case scenario if the receiving Switch ASIC waits "32 ns + 3 global_byte_clk's" from its switch_sor, then slot data from all ports will be available. The 32 ns value will translate into a certain number of global_byte_clk's once the frequency for global_byte_clk is selected. The offset will be programmable and the FIFO will be sized to accommodate the fastest possible global_byte_clk that will be allowed.

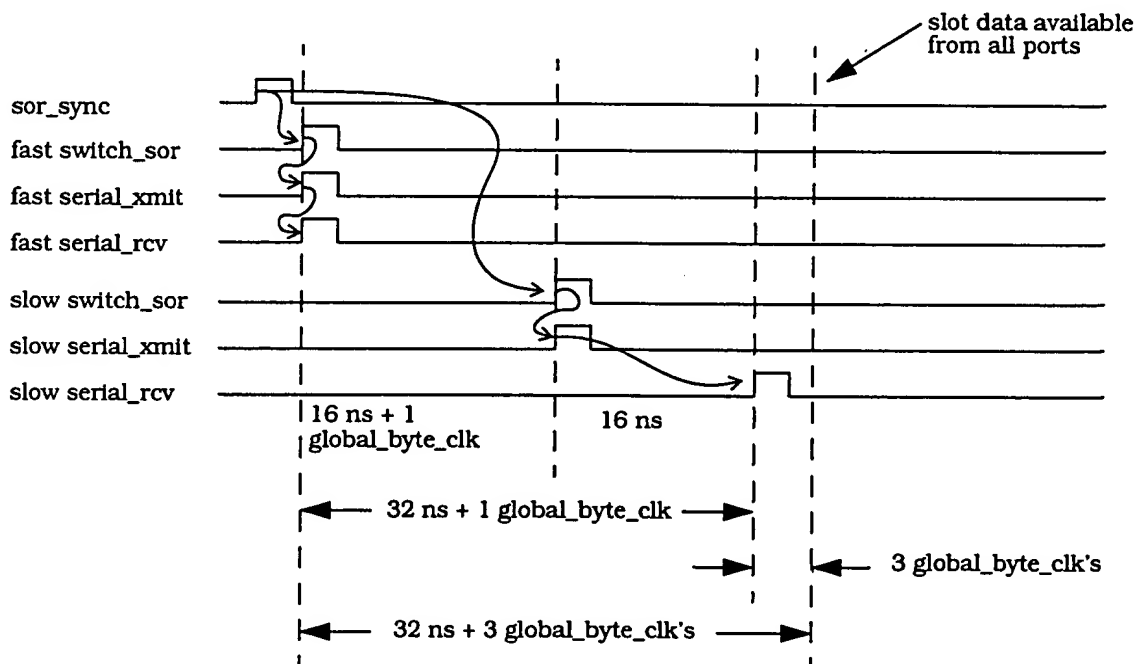


Figure 10-3: Worst Case Slot Data Available in Switch

There is also an additional consideration in determining the size of the slot data FIFO. Starting with the system layout shown in Figure 10-2, the *sor_sync* feeding the destination Switch is now the slow version as shown in Figure 10-4. Using the previous assumption that the Switch starts reading slot data at a time offset of "32 ns + 3 global_byte_clk's" from the *switch_sor*, then the slot data FIFO must be large enough to accommodate "48 ns + 4 global_byte_clk's" worth of data. This is shown in Figure 10-5. The reason for this addition requirement is because it is not know if the system

Proprietary and Confidential Information of Onex Communications Corporation

has the layout/timing properties shown in Figure 10-2 or Figure 10-4. The switch must be designed to support both scenarios.

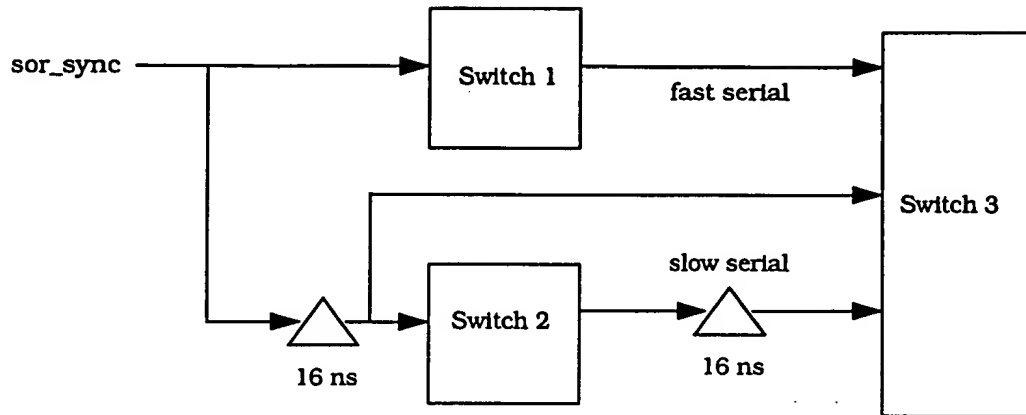


Figure 10-4: Alternate Worst Case Switch Layout Scenario

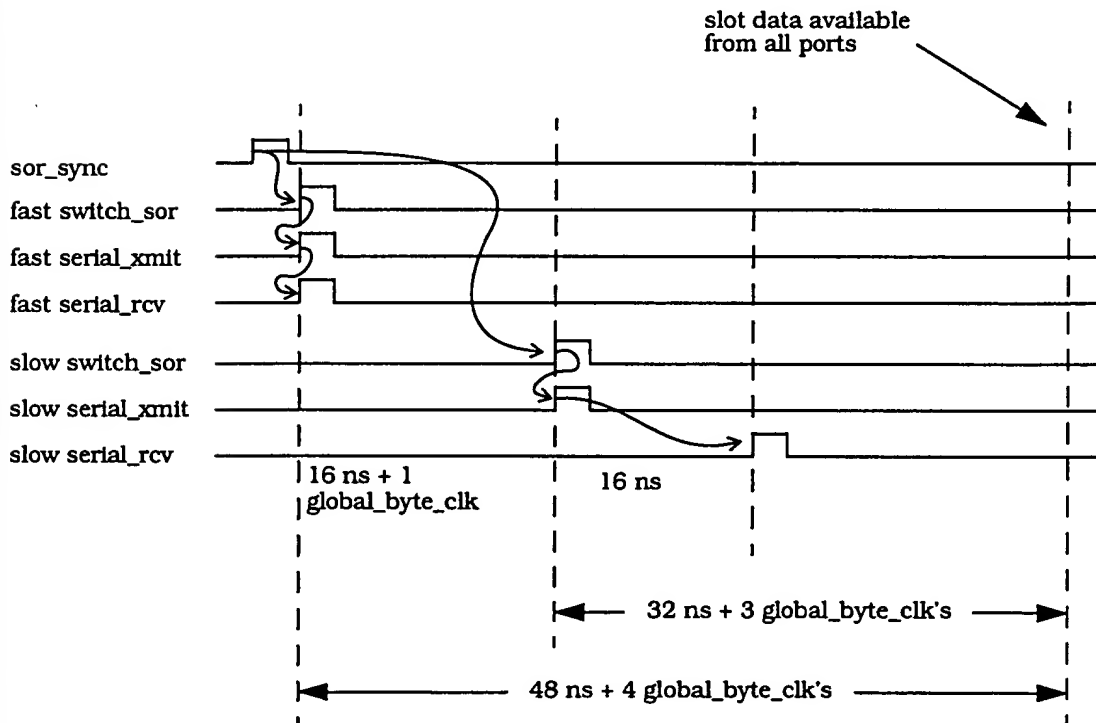


Figure 10-5: Worst Case Slot Data FIFO Requirement

To support the require slot count of 1700 slots/row, the `global_byte_clk` must operate at 275.4 MHz. This translates to a receive slot rate of 8.17 ns/slot. The `global_byte_clk` to slot ratio is 2.25 `global_byte_clk`'s per slot. Therefore, assuming the 275.4 MHz `global_byte_clk` rate, the data slot FIFO needs to be 8 slots deep based on the following:

Proprietary and Confidential Information of Onex Communications Corporation

- 48 ns => 6 slots
- 4 global_byte_clk's => 2 slots

The slot numbers are rounded up to the nearest integer. With a FIFO depth of 8 slots, the Switch can support a maximum global_byte_clk rate of 291.6 MHz which translates to 1800 slots/row.

10.4 Grant Packet Synchronization

The grant lines will synchronize in a fashion very similar to the data lines. One difference is the fact that there is only one grant input per port and therefore there is no channel synchronization required. The grant line, however, does need to pass from the link_byte_clk domain to the global_byte_clk domain. Although a three stage ring buffer could do this, the grant channels will use the same four stage ring buffer being used by the data channels (for design simplification only). The grant lines are subject to the same constraints as the data lines and therefore the grant packet FIFO must be able to buffer the same amount of data. Therefore, the grant packet FIFO will also be sized at 8 slots.

10.5 Port Processor Synchronization

As described above, all Switch ASICs receive a globally distributed sor_sync signal. This is used to synchronize their start of row timing. The Port Processor ASICs do not receive such a signal. In the iTAP system, it is possible for the Port Processor to be remotely located from the switch fabric and it would be impossible to meet the 16 ns worst case delta on the sor_sync line and the 16 ns maximum electrical delay to the Switch ASICs. The Switch ASICs, however, must still receive row data from the Port Processors that is synchronized, within the limits discussed above, to the globally distributed sor_sync line.

10.5.1 Switch ASIC Requirements

Similar to the Switch, the Port Processor has an internal counter used to maintain a 72 KHz synchronization signal. Following initialization, the Port Processor will start its 72 KHz synchronization counter and begin transmitting idle frames on its data links. As part of its standard input channel synchronization (see Section 11.2.1), the Switch input logic will synchronize to the framing pattern on the data links. Once synchronization is complete, the Switch input link(s) are now synchronized to the Port Processor start of row. There is a valid region centered about the switch_sor signal in which it is acceptable for the Port Processor start of row to be located. This is shown in Figure 10-6. If the Switch is receiving a start of row from the Port Processor within the window shown in Figure 10-6, then the Port Processor is synchronized to the Switch within acceptable limits. If not, then the Port Processor needs to be adjusted.

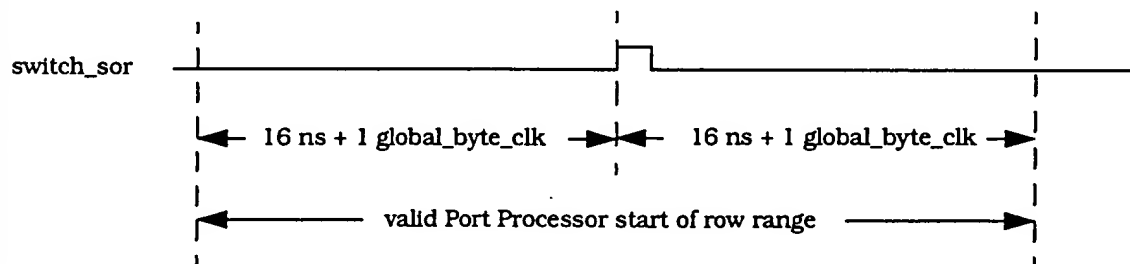


Figure 10-6: Valid Port Processor Start of Row Window

To adjust the Port Processor, the Switch must measure the difference between switch_sor and the received start of row from the Port Processor. The delta, in terms of global_byte_clk's, is always measured from the rising edge of switch_sor to the rising edge of the Port Processor start of row, as shown in Figure 10-7. The delta shown in Figure 10-7 is measured every row time. The measured delta is registered in the sync_offset_countN register (where $0 \leq N \leq 11$, one for each port) which is readable by the Tensilica processor.

Proprietary and Confidential Information of Onex Communications Corporation

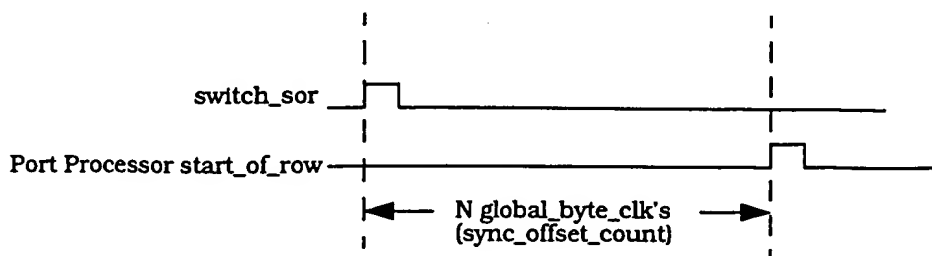


Figure 10-7: Determination of sync_offset_count

To aid with synchronizing the Port Processor, two bits are passed to the Port Processor in the link overhead field of the grant channel. These bits and their definition are summarized in Table 10-8.

Table 10-8: Port Processor Synchronization Bits

pp_synced	pp_not_synced	Comments
0	0	Switch has not determined if Port Processor is synced to Switch. This case only occurs when the Switch has not synchronized (or lost sync) to the data link.
0	1	Port Processor start of row is not within acceptable limits of switch_sor.
1	0	Port Processor start of row is within acceptable limits of switch_sor.
1	1	Invalid. Hunt the designer down like a wild animal.

Once the data link is synchronized, the Switch will check the position of the Port Processor start of row as shown in Figure 10-6. If the start of row falls within the allowable range, the pp_synced bit is set. Otherwise the pp_not_synced bit is set. In addition, the measured offset as shown in Figure 10-7 (sync_offset_count) is also inserted into the link overhead of the grant channel. (A complete definition of the link overhead field can be found in Table 11-12).

In the link overhead field of the data link, the Switch will monitor the pp_sync_done bit. This bit is set by the Port Processor once it has become synchronized to all Switches that it is connected to.

10.5.2 Port Processor Requirements

The Port Processor must be running an internal synchronization loop at a 72 KHz rate similar to what is being done in the Switch (see Section 10.1). Following a reset, the Port Processor will start its internal synchronization counter and begin transmitting idle packets with a valid framing pattern on its data links. If two ports are being used, then both ports must be transmitting the idle packets on their respective data links. The Port Processor will monitor the two bits of the link overhead section identified in Table 10-8.

If only one port is being used, then the Port Processor simply examines the pp_synced and pp_not_synced bits and determines a course of action. If the pp_synced bit is set, then the Port Processor will set the pp_sync_done bit in the link overhead field of the corresponding data link. After this, the Port Processor is good to go. If the pp_not_synced bit is set, then the Port Processor will adjust its internal synchronization based on the sync_offset_count value. This will cause the Switch to lose and then regain sync on the data link. Following this, the Port Processor should be synchronized to the Switch and the pp_synced bit should be set. If not, the adjustment procedure can be repeated. Once the Port Processor detects that the pp_synced bit is set, it must set the pp_sync_done bit.

If both ports are being used, the method to determine how to adjust the internal

Proprietary and Confidential Information of Onex Communications Corporation

synchronization counter becomes more complicated. If both ports respond with the pp_synced bits set, then the Port Processor only has to set the pp_sync_done bit and is good to go. Otherwise, the Port Processor adjust its internal synchronization. When two ports are being used, the goal is to center the Port Processor's synchronization between the two offsets received. In general, this would imply averaging the two offset values received. This method works except for the scenario shown in Figure 10-9. Based on the figure, assume the row time to be 100 clocks. Based on a random start, the Port Processor's sor falls somewhere in between the sor for switch1 and switch2. In particular, the Port Processor's sor is two clocks after switch1_sor and 4 clocks before switch2_sor. Thus the perfect adjustment for the port_processor_sor is to delay it by one clock and therefore center it between switch1_sor and switch2_sor. However, based on the fact that the sync_offset_count value is always measured from the switch_sor to the port_processor_sor, the following values are obtained:

- switch1_sor to port_processor_sor = 2
- switch2_sor to port_processor_sor = 96

Averaging these two numbers gives a result of 49, which would be an incorrect adjustment. The correct method is as follows. Based on the information shown in Figure 10-1, the largest delta between sync_offset_count's from two Switch ASICs is "32 ns + 1 global_byte_clk". Based on the global_byte_clk frequency, the 32 ns will translate into some number of global_byte_clk's. If the delta between sync_offset_count's from two Switch ASICs is greater than "32 ns + 1 global_byte_clk", then the scenario shown in Figure 10-9 has occurred. To calculate the adjustment, the row time (in this case 100) needs to be subtracted from the larger value (in this case 96). This value is then added to the smaller value and the result is averaged. Thus the algorithm is:

```

if mag(sync_offset_count1 - sync_offset_count2) > TBD programmable value
    adjust = (row_time - max(sync_offset_count1 - sync_offset_count2) +
              (min(sync_offset_count1 - sync_offset_count2))/2
else
    adjust = ave(sync_offset_count1 + sync_offset_count2)
    
```

A positive adjust implies delaying the port_processor_sor and a negative adjust implies advancing the port_processor_sor.

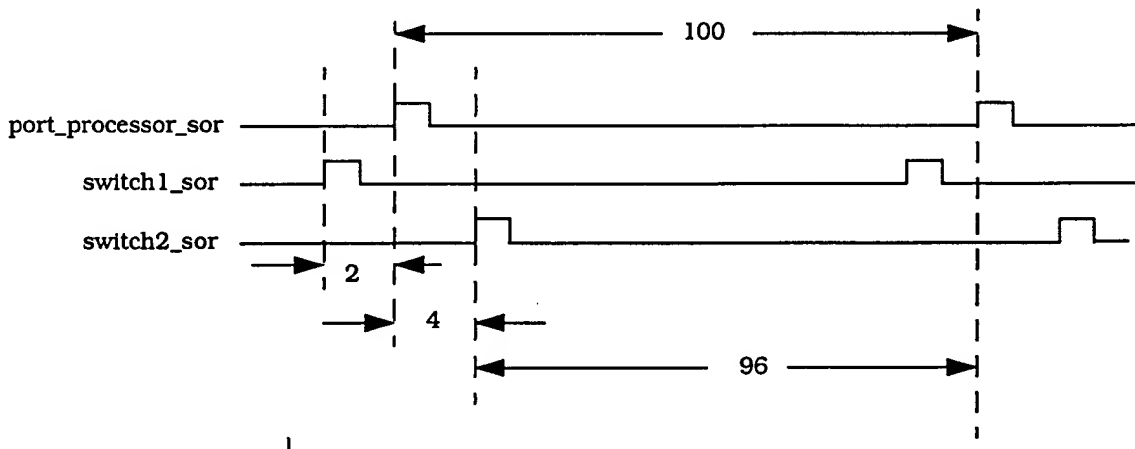


Figure 10-9: Special Case of Port Processor Synchronization Adjustment

Once the adjustment has been made, this will cause the Switch to loose and then regain sync on the data link. Following this, the Port Processor should be synchronized to both Switches and the pp_synced bit from both Switches should be set. If not, the adjustment procedure can be repeated. Once the Port Processor detects that the both pp_synced bits are set, it must set the pp_sync_done bit.

Proprietary and Confidential Information of Onex Communications Corporation

10.6 Error Monitoring

The iTAP architecture requires that all ASICs (Switches and Port Processors) remain synchronized, within the limits previously defined. Based on the fact that all ASICs are frequency locked, once the system is synchronized it should not drift out of synchronization. Provided that the individual serial channels remain synchronized (i.e. consistent framing pattern), the only way for the devices to drift out of sync with respect to each other is if their link_ref_clk's (see Section 9) are not truly frequency locked. This is a design requirement of the iTAP system and this type of problem should only arise due to some type of hardware problem. The methods to detect these types of errors are discussed below.

As mentioned in Section 10.1, once the Switch ASIC has its internal synchronization counter running, it constantly monitors the rising edge of the incoming sor_sync signal and compares it to its own switch_sor. Due to clock jitter and crossing clock boundaries, it is possible for the sor_sync signal to move slightly with respect to switch_sor. A window of +/- 2 global_byte_clk's is considered acceptable. If the rising edges of the signals vary more than this, an interrupt will be sent to the Tensilica.

The synchronization between the Port Processor and the Switch is also constantly monitored by the Port Processor. Once the Port Processor is synchronized to the Switch(es), it should never see the pp_not_synced bit set. If this occurs, it implies that the Port Processor and the Switch are not frequency locked.

Proprietary and Confidential Information of Onex Communications Corporation

11 Switch Ports

The Switch ASIC has 12 input ports and 12 output ports. Each input port to the Switch consists of 3 serial lines: 2 for data with request and one for grant. Each output port from the Switch also consists of 3 serial lines: 2 for data with request and one for grant. These lines are implemented as LVDS, Low Voltage Differential Signaling.

11.1 Serial Data Stream Format

The structure of a data/request row is shown in Figure 2-2. Splitting the row across two serial channels results in the serial data stream shown in Figure 11-1. The payload section contains 840 slots. The link overhead contains 10 slots (the contents of the link overhead field are listed in Section 11.6). Added to the end of the serial stream is a discard field containing a user programmable number of bytes that are always discarded. The required row time is 72 KHz (see Section 10) and therefore the size (in bits) of the entire serial stream will determine the clock frequency for the serial interface. The discard field allows some flexibility in the selection of the serial clock frequency.

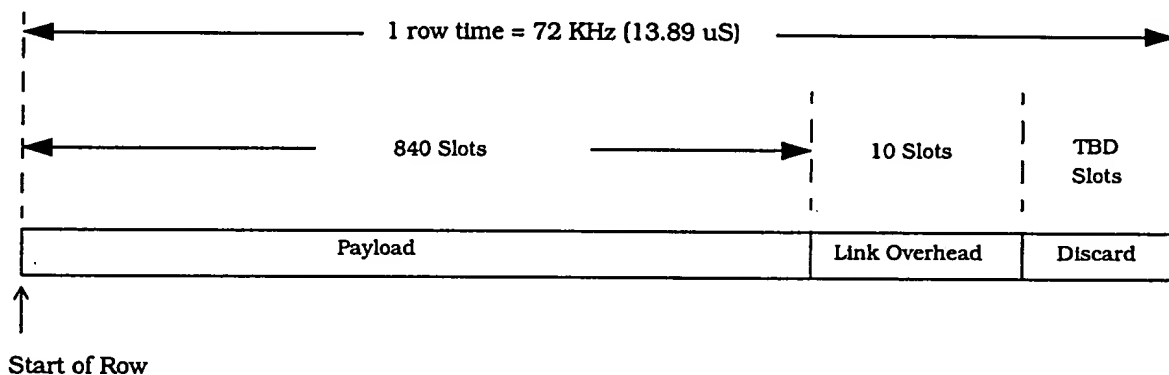


Figure 11-1: Data/Request Serial Stream Partitioning

11.2 Input Port

A top level view of the input ports is shown in Figure 11-2. A description of the signals is provided in Table 11-3. Several of the output buses are replicate N times, where $0 < N < 11$, one for each link.

Proprietary and Confidential Information of Onex Communications Corporation

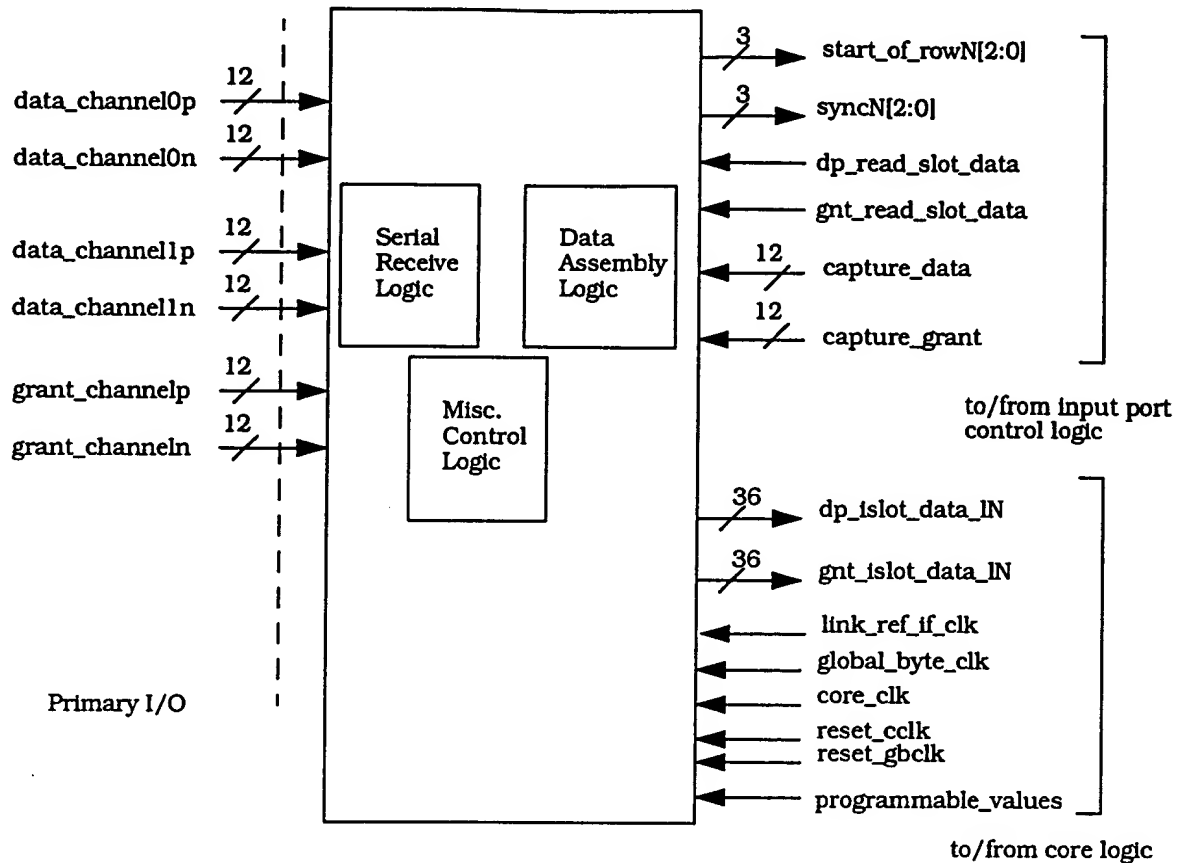


Figure 11-2: Switch Input Ports

Table 11-3: Input Port Signal Descriptions

Signal Name	Width	Direction	Description/Comments	Source/Dest	Active State
data_channel0-1p(n)		input	Differential serial lines used for data. Data and request are parsed across two channels to achieve the necessary 3.8 Gbps bandwidth.		NA
grant_channel1p(n)		input	Differential serial line used for the incoming grant information.		NA
start_of_row[2:0]		output	When active, indicates the receipt of the start of row on the corresponding serial channel.		High
sync[2:0]		output	When active, indicates that the corresponding serial channel is synchronized.		High
read_slot_data		input	A data slot will be read from the input port for every core_clock in which read_slot_data is high.		High
read_grant_data		input	A grant packet will be read from the input port for every core_clock in which read_grant_data is high.		High
capture_data		input	When this signal is active, at the start of the next frame all data channel input FIFOs (associated with the respective port) will start writing data and continue to write data until the signal is inactive.		High

Table 11-3: Input Port Signal Descriptions

Signal Name	Width	Direction	Description/Comments	Source/ Dest	Active State
capture_grant		Input	When this signal is active, at the start of the next frame the grant channel input FIFO will start writing data and continue to write data until the signal is inactive.		High
link_ref_if_clk	1	Input	Clock used by the deserializer/clock recovery logic (see Section 9).		NA
global_byte_clk	1	Input	Clock used by the port logic (see Section 9).		NA
core_clk	1	Input	Clock used by the core logic (see Section 10).		NA
reset_cclk	1	Input	ASIC reset, synchronized to the core clock. When active, all registers are synchronously placed into a know state. Reset must remain active for a minimum of TBD byte clocks.		Low
reset_gbclk	1	Input	ASIC reset, synchronized to the global byte clock. When active, all registers are synchronously placed into a know state. Reset must remain active for a minimum of TBD byte clocks.		Low
programmable_values		input/ output	The input port has several programmable registers as defined in Table 11-11.		NA

11.2.1 High Speed Serial Input Logic Blocks

All high speed serial inputs have a common front end as shown in Figure 11-4. Basic operation is as follows. Serial data is converted to byte wide data and then passed through a descrambler to a four stage ring buffer. The ring buffer is used to compensate for differential arrival times of the serial data across two serial lines and to provide the ability to transfer across clock domains. The byte data is also passed to frame detection logic that finds a framing pattern and provides bit shift information to the deserializer. Prior to being used to communicate data, each line must be bit aligned at the receiver. Once this is done, the receiving logic must be synchronized to a framing pattern. After completing this, the serial link is considered synchronized.

There are four different clock domains that are associated with the port logic: core_clk, global_byte_clk, link_byte_clk, and link_serial_clk. Detailed information about these clocks can be found in Section 9. Referring to Figure 11-4, the link_serial_clk is used in the deserializer and clock recovery logic. The output side of the ring buffer is operated in the global_byte_clk domain. All other logic shown in Figure 11-4 is operated in the link_byte_clk domain.

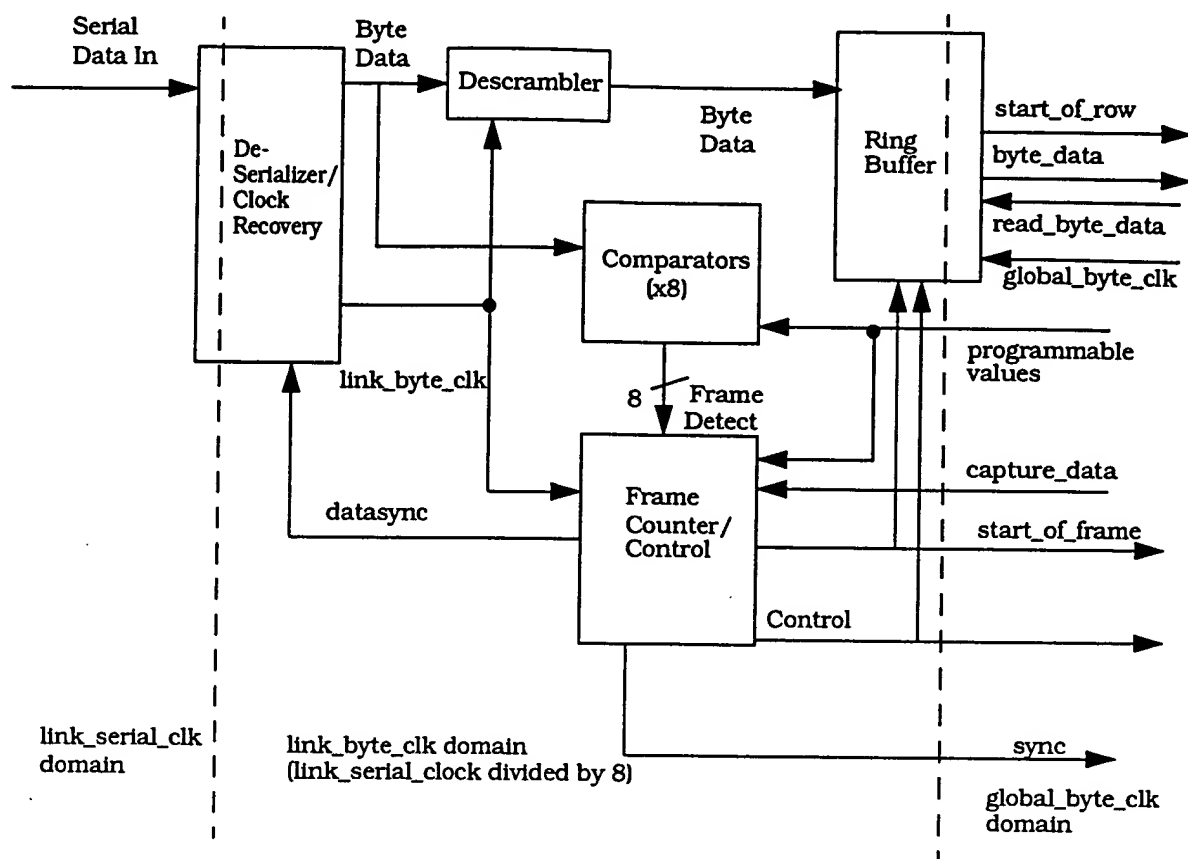


Figure 11-4: High Speed Serial Input Functional Block Diagram

11.2.1.1 Deserializer

Serial data enters a high speed deserializer. In this logic, a clock recovery of the transmitting clock is made and the serial data is converted to a byte wide output. The byte wide data is made available as well as link_byte_clk ($/8$ of the recovered clock). The logic also has an input, datasync , that allows a one bit shift of the output byte data every time the bit_shift line is toggled. This logic will be provided as IP by the ASIC vendor.

11.2.1.2 Descrambler

The descrambler uses a $1+x^6+x^7$ polynomial identical to the one used for SONET. All serial data is scrambled using a corresponding scrambler before being transmitted. The descrambler is reset to its initial seed value of all 1's at the start of every frame. As is shown in Figure 11-4, data is taken directly from the deserializer block to the Comparators block (used to detect the framing pattern). The two byte framing pattern is not scrambled on the transmit side and therefore it is not required to be de-scrambled on the receiving side. All other data is scrambled and must be descrambled before being written to the FIFO.

11.2.1.3 Comparators

The comparators compare the byte data to a programmable two byte framing pattern and provide a frame_detect output. To improve the speed at which framing synchronization can be made or re-established when lost, there are actually 8 parallel comparators. Each comparator is comparing the framing pattern to one of 8 possible shifted versions of the input data. Based on the comparator that detects the framing pattern, the deserializer can be instructed to shift the appropriate number of bits.

11.2.1.4 Frame Counters and Control

The frame counter is used in conjunction with the frame detect to detect and ensure a correct framing pattern. Once a framing pattern has been detected, the frame counter will check to ensure that the framing pattern is detected in the same position of each incoming row of data. After receiving a programmable number of consecutive correctly placed framing patterns, the sync line will be set indicating that the interface is synchronized. Once the interface is considered synchronized, the start_of_row output will be pulsed at the start of each row of data. Any frame of data in which the framing pattern is not correctly detected will result in the generation of a framing_error. After a programmable number of consecutive framing errors, the sync line is reset and the interface is now considered not synchronized. At this time, the interface will attempt to re-establish sync.

In addition to data, a start_of_row bit is also written to the ring buffer. The start_of_row bit will be set high to coincide with the first byte of data in each row.

11.2.1.5 Ring Buffer

The ring buffer serves two purposes. First, it allows the two data serial lines to be synchronized at the byte level. The necessity for this is discussed below. Second, it provides a method to cleanly transfer data across clock boundaries. The write side frequency is operating from a link_byte_clk clock. The read side is operating at the same frequency, but from the global_byte_clk clock. There is no phase relationships between the clocks, but they are frequency locked such that overrun/underrun conditions will not occur within the ring buffer.

The ring buffer is implemented using four stages. Two stages are required to cleanly transfer data from the link_byte_clk domain to the global_byte_clk domain. A third stage provides buffering to allow a one link_byte_clk clock difference between the receipt of serial data on the two serial lines comprising the data link. This is required because on the transmitting side, the link_byte_clk's are frequency locked but not phase locked. A fourth stage is added to compensate for electrical deltas between the two serial lines comprising the data link. Assuming a byte rate of 275.4 MHz, an electrical delta of 3.6 ns can be compensated for with the additional buffer stage.

11.2.2 Input Data Line Grouping

To support its maximum capability, the iTAP architecture requires a bandwidth of 3.8 Gbps on each data/request port.

Data/request is always presented to the core as a 36 bit data slot plus an islot_start_of_row control bit. The islot_start_of_row control bit will be high when the corresponding slot is the first slot of the row. Otherwise, islot_start_of_row will be low. To assemble the byte data from the two serial channels, one byte is read from each ring buffer and concatenated into an 16 bit wide word with serial channel 0 always as the MSB. Reading from the ring buffer continues in this manner with the new 16 bit word being concatenated to the LSB of the previous word. The concatenated data is then parsed into 36 bit slots and stored in a FIFO. A separate top level control module (see Section 11.3) will control the reading of the slot data.

The start of row control bit from all data channels is compared to ensure that all channels are aligned. If there is a discrepancy, an error flag is set and the misalignment counter for the corresponding data channel is incremented. The data misalignment counter is three bits and will stop incrementing once it has reached its maximum value. An error counter size of only three bits is based on the fact that a start of row misalignment can only be caused by the data channel losing sync or a spurious glitch in the data assembly logic. In the first case, losing sync implies that the channel is down, all data is voided, and the channel will be reset once sync is re-established. In the latter case, a single error over an extended period might be in the realm of possibilities. However, several errors in a row probably indicates that the data assembly logic is out of sync and needs to be reset.

As was shown in Figure 11-4, the write side of the ring buffer is in the link_byte_clk domain. The read side of the data channel assembly logic operates in the core_clk domain. All other logic associated with Figure 11-5 operates in the global_byte_clk domain.

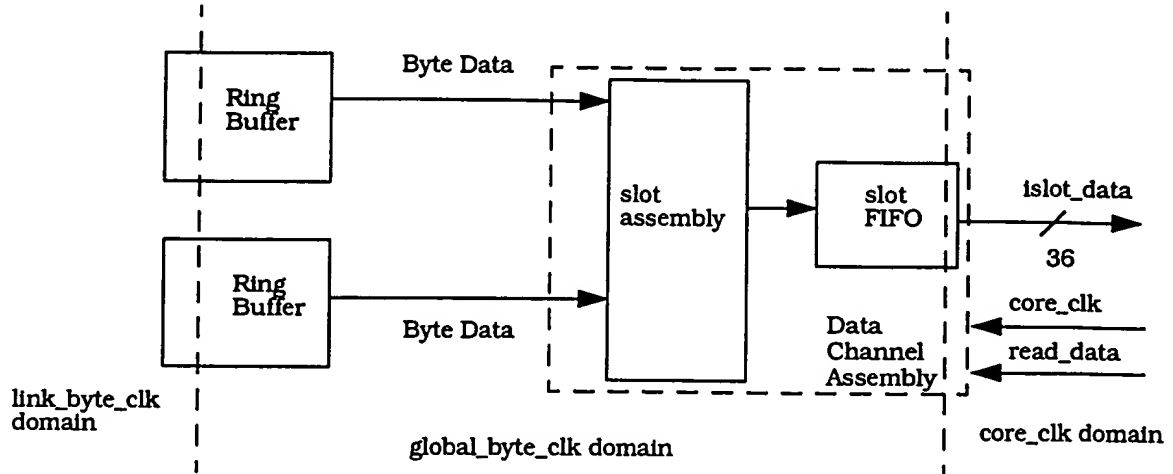


Figure 11-5: Channel Assembly for Serial Data Channels

11.2.2.1 Data Slot FIFO

The slot FIFO serves two functions: transferring slot data from the *global_byte_clk* domain to the *core_clk* domain and provide buffering to allow slot alignment of all input ports. As documented in Section 10.3, the slot FIFO should be sized at a minimum of 8 slots. The read and write domains of the slot FIFO are totally asynchronous and have no relationship to each other. Based on the clocking requirements (see Section 9) it is guaranteed that the read rate is faster than the write rate. Therefore, the FIFO logic does not need to support a "full" condition. To ensure correct operation, Johnston counters will be used to monitor read and write pointers.

11.2.3 Grant Packets

The grant channel is handled in a similar manner to the data channels. The main difference is that the grant interface consists of only one serial channel. Therefore, the slot assembly only requires assembling and partitioning data from one serial stream. As a result of this, the grant slot FIFO needs to only be 4 slots deep (see Section 10.4). The read and write domains of the slot FIFO are totally asynchronous and have no relationship to each other. Based on the clocking requirements (see Section 9) it is guaranteed that the read rate is faster than the write rate. Therefore, the FIFO logic does not need to support a "full" condition. To ensure correct operation, Johnston counters will be used to monitor read and write pointers.

11.3 Input Port Control

To support the iTAP architecture, the core logic must receive input data that is synchronized across all 12 input ports (data and grant). To achieve this, a top level input port controller is used to synchronize events across all input ports (the synchronization methodology is described in detail in Section 10). A top level view of the input port control is shown in Figure 11-6. A description of the signals that interface to the core is provided in Table 11-7. A description of the signals that interface to the input port logic is provided in Table 11-3.

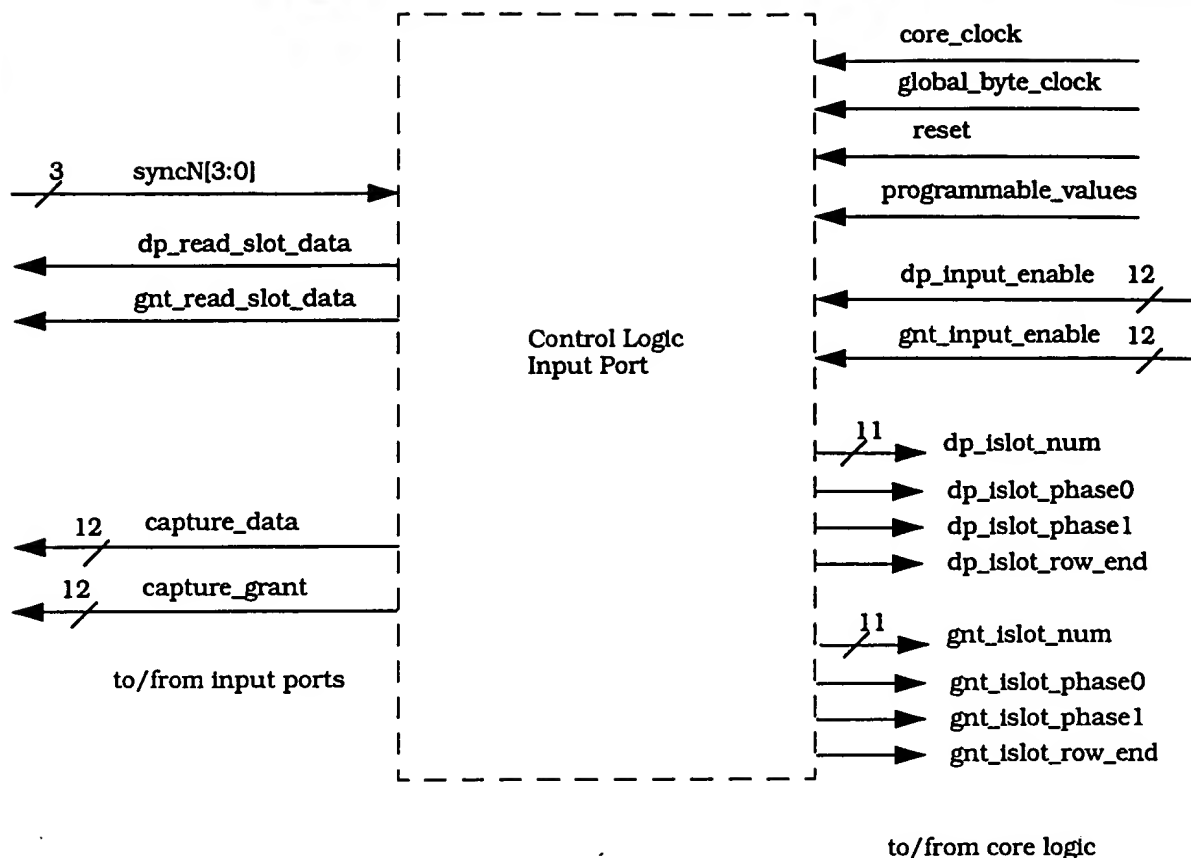


Figure 11-6: Top Level View of the Input Port Control Logic

Table 11-7: Input Port Control Logic Signal Descriptions

Signal Name	Width	Direction	Description/Comments	Source/ Dest	Active State
port_clock			Clock used by the port logic (see Section 9).		NA
core_clock			Clock used by the core logic (see Section 9).		NA
reset			ASIC reset. When active, all registers are synchronously placed into a know state. Reset must remain active for a minimum of TBD byte clocks.		Low
idslot_num	11	output	Indicates the current input data slot number.		
idslot_phase0	1	output	This signal is asserted during the first cclk cycle for each new incoming data slot period.		High
idslot_phase1	1	output	This signal is asserted during the second cclk cycle for each new incoming data slot period.		High
idslot_data	36	output	Current input data slot contents		NA

Table 11-7: Input Port Control Logic Signal Descriptions

Signal Name	Width	Direction	Description/Comments	Source/ Dest	Active State
idslot_row_end	1	output	This input is asserted during the last data slot of the row. If the serial input links are configured to carry more than 1700 slots per row, then this input will be asserted for slots 1699 and higher. Speeding up the input links will only be done as part of the IC characterization, not intended for normal operation.		High
igslot_num	11	output	Indicates the current input grant slot number.		
igslot_phase0	1	output	This signal is asserted during the first cclk cycle for each new incoming grant slot period.		High
igslot_phase1	1	output	This signal is asserted during the second cclk cycle for each new incoming grant slot period.		High
igslot_data	36	output	Current input grant slot contents		NA
igslot_row_end	1	output	This input is asserted during the last grant slot of the row. If the serial input links are configured to carry more than 850 slots per row, then this input will be asserted for slots 849 and higher. Speeding up the input links will only be done as part of the IC characterization, not intended for normal operation.		High
sor_sync			Primary input sor_sync.		NA
switch_sor			Internal synchronization signal (see Section 10.1).		NA
programmable_values			The input port has several programmable registers as defined in Table 11-11.		NA
valid_data			For every core_clock in which valid_data is high, the output slot_data contains valid slot data and must be latched in the current core_clock cycle.		High
valid_grant			For every core_clock in which valid_grant is high, the output grant_data contains a valid grant packet and must be latched in the current core_clock cycle.		High
core_ready_for_data			When active, this signal indicates that the core is ready to receive slot data.		High
core_ready_for_grant			When active, this signal indicates that the core is ready to receive grant packets.		High
error_information			Error information available to the core as described in		NA
select_error_information			Selects which error information is made available to the core.		NA

11.4 Output Ports

A top level view of the output ports is shown in Figure 11-8. A description of the signals is provided in Table 11-9.

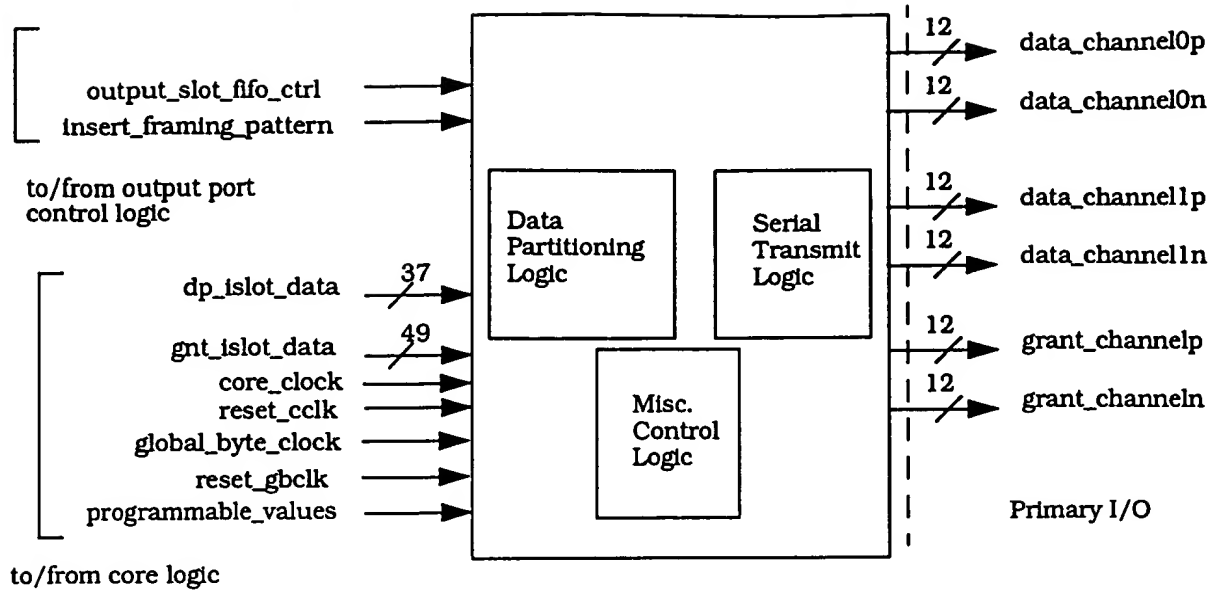


Figure 11-8: Switch Output Port

Table 11-9: Output Port Signal Descriptions

Signal Name	Description/Comments	Active State
data_channel0-1	Differential serial lines used for data. Data and request are parsed across both channels to achieve the necessary 3.8 Gbps bandwidth.	NA
grant	Differential serial line used for the out going grant information.	NA
data_source	Selects if the data transmitted on the serial data/request channels will be slot data (=1) or the idle pattern (=0).	NA
grant_source	Selects if the data transmitted on the grant channel will be grant data (=1) or the idle pattern (=0).	NA
insert_framing_pattern	When this signal is active, the user programmable framing pattern will be inserted into all the output serial channels.	High
load_data	When this signal is active, the output port logic will latch in the current value on the slot_data lines.	High
load_grant	When this signal is active, the output port logic will latch in the current value on the grant_data lines.	High
serial_clock	Clock used by the deserializer/clock recovery logic (see Section 9).	NA
port_clock	Clock used by the port logic (see Section 9).	
core_clock	Clock used by the core logic (see Section 9).	
reset	ASIC reset. When active, all registers are synchronously placed into a know state. Reset must remain active for a minimum of TBD byte clocks.	Low
slot_data[37:0]	The 36 LSBs ([35:0]) contain the out going slot data. The MSB ([35]) is high when the corresponding slot data contains the start of row.	NA
grant_data[48:0]	The 48 LSBs ([47:0]) contain an out going grant packet. The MSB ([48]) is high when the corresponding grant packet contains the start of row.	NA
programmable_values	The input port has several programmable registers as defined in Table 11-11.	NA

11.4.1 High Speed Serial Output Logic Blocks

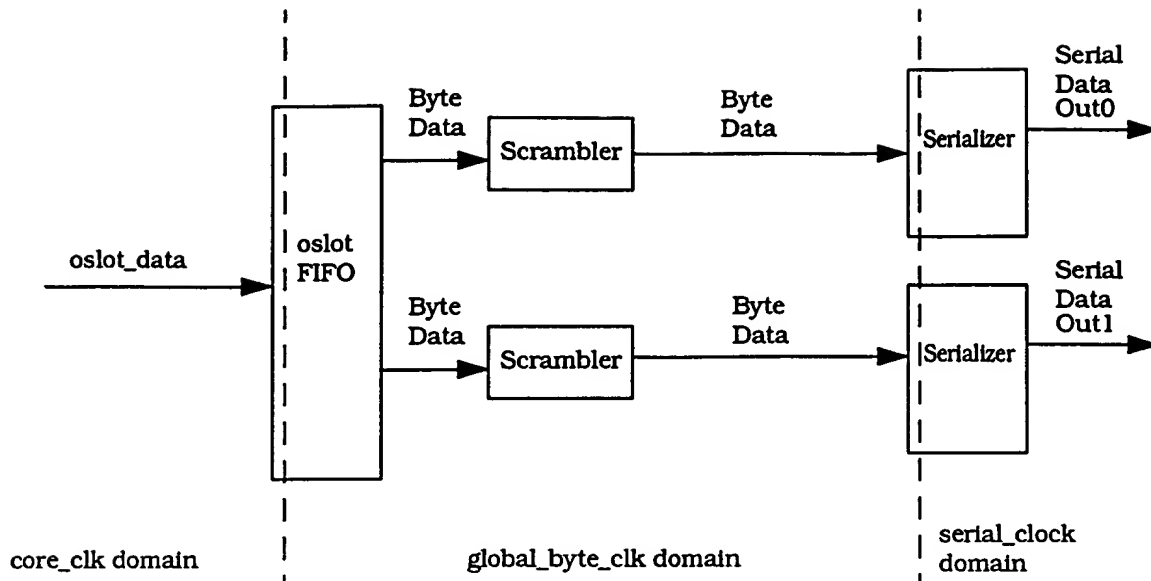


Figure 11-10: High Speed Serial Output Functional Block Diagram

11.4.1.1 Serializer

This logic will be provided as IP by the ASIC vendor.

11.4.1.2 Scrambler

The scrambler uses a $1+x^6+x^7$ polynomial identical to the one used for SONET. The scrambler is reset to its initial seed value of all 1's at the start of every row. The scrambler function is bypassed for the two bytes containing the framing pattern (i.e. the framing pattern is not scrambled).

11.4.1.3 Link Overhead Buffer

The link overhead buffer provides the source for the link overhead section of the serial data stream (see Section 11.1). The buffer can be sized at a maximum of 19 slot which are all accessible to the Tensilica processor. Currently the buffer is sized at TBD slots. Table 11-11 and Table 11-12 identify the currently defined usage of the link overhead.

11.5 Output Port Control

11.6 Link Overhead Information

As shown in Figure 11-1, all serial channels have an associated link overhead section. This information is discussed in Section 4.3.1.5.3. Because the link overhead data is not switched, it can only be used as a method to communicate between neighboring devices. The I/O logic makes available the synchronization status of each input serial line to be placed into the link overhead. This allows neighboring ASICs to know the synchronization status of their respective outgoing serial channels.

In addition, the sync_offset_count and sync_offset_count_valid values are made available to be inserted into the grant channel link overhead. More information about these fields can be found in Section 10.5.

11.7 Programmable Registers

Each Switch port has associated with it the user programmable registers identified in Table 11-11. The Port/Chip column indicates if there is one register per port (P) or one register per chip (C).

Table 11-11: Switch Port Programmable Registers

Name	Size (bits)	Default	Description
frame_size	11		Size of a row in bytes for the serial data channels
frame_pattern	16	0xF628	The value to be used as the framing pattern
lostsync_loop_limit	4	0xa	The number of consecutive framing errors required before a synchronized serial channel is considered out of sync
pre_sync_count	4	0x3	The number of consecutive correct framing patterns required before a serial channel is considered synchronized
gate_capture_data_position	13	0x30	Set to a value that ensures that the start of row from all inputs has been received
total_data_slot_limit	11		Set to one less than the total number of data slots in a row
valid_data_slot_limit	11		Set to two less than the total number of valid data slots in a row
data_row_toggle_position	11		Set to one less than the slot number in which dp_data_row_toggle should toggle
total_grant_slot_limit	11		Set to one less than the total number of grant slots in a row
valid_grant_slot_limit	11		Set to two less than the total number of valid grant slots in a row
grant_row_toggle_position	11		Set to one less than the slot number in which gnt_data_row_toggle should toggle
data_framing_pattern_position	13		Set = (position of the first byte of the framing pattern in the out going byte stream + start_output_position + 4).
grant_framing_pattern_position	13		Set = (position of the first byte of the framing pattern in the out going byte stream + start_output_position + 4).
data_sor_offset	13		Set = (frame_size - data_framing_pattern_position + 2)
grant_sor_offset	13		Set = (frame_size - grant_framing_pattern_position + 2)
start_output_position	13	0x1	Instructs the switch when to start outputting data. The default value of one should be used unless lab testing is being done.
start_slot_position	13		Set = (frame_size - 10) for normal operation. This value must be changed if the core clock and byte clock ratio are changed for testing.
data_link_input_enable	12	0x0	Active high to enable input data links. One bit per link.
grant_link_input_enable	12	0x0	Active high to enable input grant links. One bit per link.
data_link_output_enable	12	0x0	Active high to enable output data links. One bit per link.
grant_link_output_enable	12	0x0	Active high to enable output grant links. One bit per link.
disable_scrambler	1	0	Active high to disable the scrambling function. Should only be used for Verilog simulations.
disable_descrambler	1	0	Active high to disable the descrambling function. Should only be used for Verilog simulations.
bypass_sync	1	0	Active high to bypass the synchronization loop counters. A serial channel will be synchronized as soon as it detects the framing pattern. Should only be used for Verilog simulations.
switch_sor_reset	1	0	Active high to cause the internal synchronization engine to re-synchronize (see Section 10.1).

Proprietary and Confidential Information of Onex Communications Corporation

Table 11-11: Switch Port Programmable Registers

Name	Size (bits)	Default	Description
slot_size_select	2	0x0	Sets the slot size for the serial links. Only used for testing with different serial frequencies. 00 - 36 bits 01 - 40 bits 10 - 44 bits 11 - 48 bits

11.8 Core Interface Timing

The interface timing between the core and the input logic is documented in Section 4.3.1.2.

100-00111-00.111

Proprietary and Confidential Information of Onex Communications Corporation

12 RISC Controller

SUBJECT TO TENSILICA NDA

1111_1111.111

13 Sbus Modules

13.1 SPI Interface

The SPI interface is a 4 wire serial interface developed by Motorola for low speed synchronous serial communications. For the iTAP chipset, the SPI interface will be used to connect a serial eeprom to the system for purposes of loading the local microprocessors as well as for storing certain state information in non-volatile memory.

The iTAP SPI module supports the following features:

- Configurable Speed settings- 500KHz, 1MHz, 2MHz and 4MHz (w/ nominal 250MHz clock)
- Automatic Page Buffer Management
- Configurable Page Buffer Size
- 2 SPI Devices (i.e. 2 chip selects)
- Interrupt Driven 'Request' Based mode.
- 16 and 24 Bit Address Mode Compatible

The SPI bus developed by Motorola allows for 4 different modes of operation. They are distinguished by when the clock starts to toggle with respect to a peripheral's chip select and which edge of the clock to transmit/receive data on. In the Motorola SPI these are governed by the CPHA and CPOL bits. Out of these 4 modes, vendors (Atmel, Xicor, Microchip) for serial eeprom devices support modes 0 and 3 (0,0 & 1,1). In both of these modes, data is transmitted on the negative edge of the clock and received on the positive edge. The difference in the modes is that the clock in mode 0 should be low when cs is active before the transfer begins, mode 3 allows for a negative edge transition of the clock. A mode 0 compatible boot loader will be sufficient. If a mode 1 or 2 device needs to be hooked up, an external inverter will be needed on the clock line. In addition, vendors of the proms have developed faster versions of the chips which aren't bound to the original 2MHz specification.

13.1.1 Theory of Operations

The iTAP SPI has 2 modes of operation, normal and interrupt driven. Both are discussed below.

13.1.1.1 Normal Mode

In 'Normal Mode' of operation the SBUS master can perform reads and writes on the SBUS by accessing memory from 0x4000 to 0x400000. This allows for a 4 megabyte external device to be supported. In this mode, the SPI looks like a chunk of memory which is just very slow. Behind the scenes, the SPI controller takes care of enabling the EEPROM for writes and reading the SPI status register. Because of this, the Tensilica can have its reset vector in the SPI space, and boot out of it just as it would from any memory.

When a Read is performed, the SPI controller will check to see if it has an open page, and if so close it. Then it will check the Status register to see if it is possible to perform a read. When the status register acknowledge the device is ready for an access, the read is performed, and the data transferred to the SBUS master. The time that it takes for this access could be as great as 12 ms, which is the time it takes for an SPI program cycle.

When a Write is performed, the SPI controller will check to see if a 'page' is currently being written, and to see if the address matches that of the current page. If it is, the data is written and the cycle terminates. Otherwise, the page needs to be closed, program initiated, the status register polled, and then the data written to the SPI. This could be as long as 12ms. The bus cycle ends as soon as the data is written into the SPI, but before a programming cycle is initiated to speed up transactions.

13.1.1.2 Interrupt Driven Mode

In this mode, the SBUS master will write to an address register, a data register (if a write) and a command register. This will initiate an SPI access. When the access is complete an interrupt will be signaled on. It is also possible to poll a Done bit to see if the transaction has been completed.

The purpose of this mode is for the host interface to program the SPI and not need to have an impossibly long bus time-out value. Otherwise, each individual bus access would have a minimum time of several microseconds to 'dittle' the SPI bus and 12 ms to actually program a page into the SPI. In an iTAP system, the bus timeout will be much shorter (on the order of several dozen clock ticks

Proprietary and Confidential Information of Onex Communications Corporation

~100ns).

13.1.1.3 SPI Settings

- **SPI Serial Clock Speed**- a 2 bit field to control the speed of the clock as a function of the core clock. With a nominal 250MHz core clock speed combinations from 500KHz to 4MHz are possible.
- **SPI Addressing Mode**- a single bit to control whether or not 16 or 24 bit addressing is used for the SPI prom. For the 64Kbyte and under devices, they need to have 16 bits written into them as address. For the larger devices, they need to have 24 bits written into their address buffers. The reset state of the register is determined by bootmode[2].
- **SPI Page Register Size**- an 8 bit field to determine the number of consecutive addressed bytes that can be written into the SPI before the page has to be closed for writing. The reset value of this register is zero, since some small devices do not have a page register at all.
- **SPI CS**- a single bit to determine which of the SPI chip selects will be active. The reset state of this register is determined by the bootmode[0] pin.

13.1.2 SPI Memory Map and Registers

Address	Name	R/w	Size
0x0000	SPI Status Register	R/W	8
0x0004	SPI Write Enable Latch	W	8
0x0008	SPI Write DI Latch	W	8
0x000C	SPI Address	R/W	32
0x0010	SPI Data	R/W	32
0x0014	SPI Stat	R/W	32
0x4000-	SPI Memory	R/W	32

13.1.2.1 SPI Status Register

Bit	7	6	5	4	3	2	1	0
Field	WPEN	0	0	0	BPI1	BPI0	WEN	RDYx

Consult SPI data sheet for a more complete description, as the BPI fields are part-specific.

13.1.2.2 SPI Write Enable Latch

Bit	7	6	5	4	3	2	1	0
Field	x	x	x	x	x	x	x	x

Accesses to this address will generate a Write Enable Command on the SPI.

13.1.2.3 SPI Reset Write Enable Latch

Bit	7	6	5	4	3	2	1	0
Field	x	x	x	x	x	x	x	x

Accesses to this address will generate a write disable command on the SPI.

13.1.2.4 SPI Address (Interrupt Mode)

This is a read/write 32 bit register. The upper 4 bits control whether or not its a read or write. Setting

Bit	31	30	29	28	27	26	25	24	23	0
-----	----	----	----	----	----	----	----	----	----	---

Field	BWE3	BWE2	BWE1	BWE0	0	0	0	0	Address
-------	------	------	------	------	---	---	---	---	---------

To perform a read, program BWE[3:0] = 0, for a write operation set the bits which should be programmed. The following settings are allowed:

BWE3	BWE2	BWE1	BWE0	Access
0	0	0	0	Read
1	0	0	0	Write bits [31:24]
0	1	0	0	Write bits [23:16]
0	0	1	0	Write bits [15:8]
0	0	0	1	Write bits [7:0]
1	1	0	0	Write bits [31:16]
0	0	1	1	Write bits [15:0]
1	1	1	1	Write bits [31:0]

13.1.2.5 SPI Data (Interrupt Mode)

This is a read/write 32 bit register. Data which is to be written to the SPI should be written into here. Data which is to be read will be read here.

13.1.2.6 SPI Stat (Interrupt Mode)

This register controls the interrupt based mode transfers to the SPI.

Bit	7	6	5	4	3	2	1	0
Field	Go	0	0	0	0	0	0	0

Writes to the GO bit will initiate a SPI bus cycle. During read cycles, if this bit is set, the SPI access is in progress. An interrupt will be generated on the hi to low transition of this bit. This interrupt may be masked via the interrupt controller. The interrupt bit number is TBD.

13.1.2.7 MReset Register Applicable Bits

The control bits for the spi register are located in one of the bytes of the MRESET register. (Master Reset Register), which is located at 0x45038.

31		28		27		24		23		20		19		16		15		12		11		8		7		0		Address Offset								
Module Resets										SPI Page Size										SPI A	SPI_CS	SPI Mode	reserved				BootMode						XT CLK	WRST	0	
0	0	0	0	0	0	0	0	0	0									BootMode[2]	BootMode[0]	0	0						0	0	0	0	0	0	0	0	1	Reset Value

SPI_A: Set hi for 24 bit addressing, low for 16 bit addressing. Consult individual SPI EEPROM data sheet for the correct mode.

SPI_CS: Set hi for CS0, low for CS1.

SPI_MODE: Controls the SPI toggle clock speed (assuming 250MHz nominal core clock).

00 - 500KHz

01 - 1MHz

10 - 2MHz

Proprietary and Confidential Information of Onex Communications Corporation

for diagnostical or debug reasons.

The loopback and daisychain shunt will each have a bit to control whether or not the uart is in a particular mode.

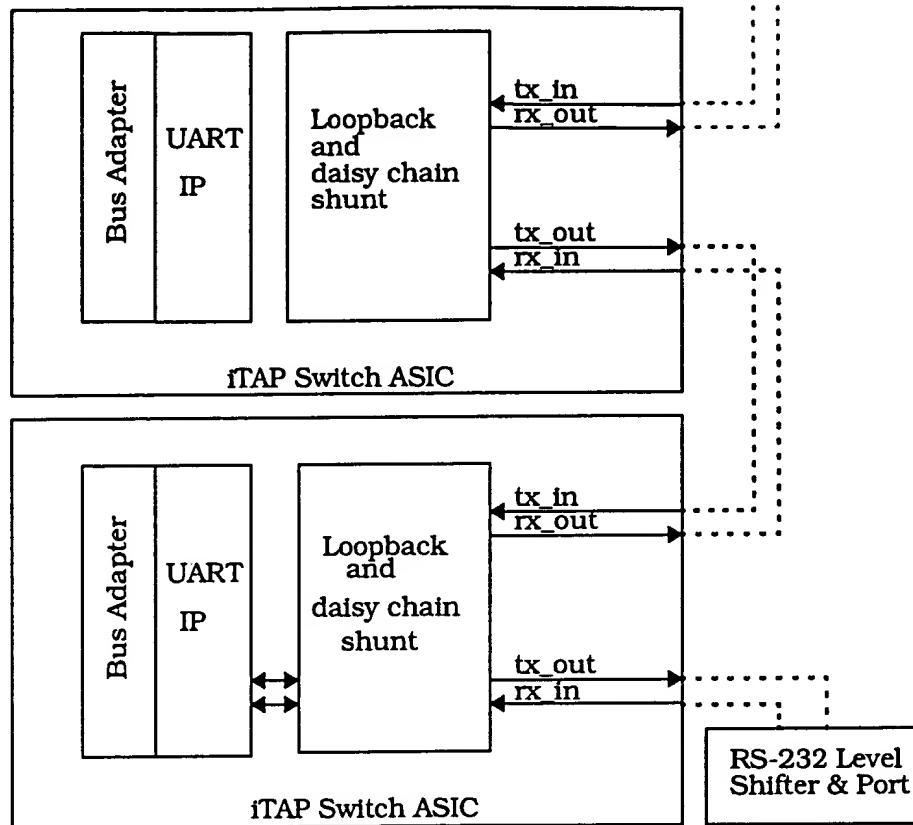


Figure 13-1: Daisy Chained Serial Interface

13.3 iTAP Switch Configuration Registers

In addition to these peripherals there will be many configuration registers which are also accessible through the SBus. These are described below.

Base Address 0x00440000								
31	24	23	16	15	8	7	0	Address Offset
iLink Control 00								0x0000
iLink Control 01								
iLink Control 02								
iLink Control 03								
iLink Control 04								
iLink Control 05								
iLink Control 06								
iLink Control 07								
iLink Control 08								

Proprietary and Confidential Information of Onex Communications Corporation

the 12 output links. Bit 27 refers to Link 11, Bit 16 refers to Link 0. The reset value is all 1's so that any input is allowed to goto every output without generating an error condition.

Stage Number: This 3 bit field sets the stage number for the iTAP Switch Element Link. The reset value is 0.

13.3.2 iTAP Switch ID Register

31	28	27	24	23	20	19	16	15	12	11	8	7	0	Address Offset
Switch ID												reserved		0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	Reset Value

This register will hold the individual switch identification number. This number shall be unique to the switch matrix, and programmable by the internal master processor or through the host interface. This ID number is transmitted in one of the Link Overhead fields.

13.3.3 iTAP Switch Part Number

31	28	27	24	23	20	19	16	15	12	11	8	7	0	Address Offset
Switch Part Number												Switch Revision		0
TBD												0	0	Reset Value

This is a 32 bit read only register which contains the switch revision and part number. This will allow multiple generations of devices to share the common code base.

This value is TBD

13.3.4 iTAP Switch Reset Register

31	28	27	24	23	20	19	16	15	12	11	8	7	0	Address Offset
Module Resets								SPI Page Size				SPI A	SPI CS	0
0	0	0	0	0	0	0	0							Reset Value

Module Resets: Write a logical 1 to any of these bits to bring that module out of reset.

Bit 31: Master Processor

Bit 30: DataPath

Bit 29: Arbitration

Bit 28: MultiCast

Bit 27: Serial / Deserializer

Bit 26: unassigned

Bit 25: unassigned

Bit 24: unassigned

SPI Page Size, SPI_A, SPI_CS, SPI_MODE: All of these bits are described in the SPI section of this document.

Proprietary and Confidential Information of Onex Communications Corporation

13.3.5 iTAP Bus Control

31	28	27	24	23	20	19	16	15	12	11	8	7	0	Address Offset		
Full Decode								Bus Timer								0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	Reset Value		

Full Decode: Setting these bits force the target module to fully decode the address bus. Otherwise, the decoded module should never bus error during unmapped accesses.

Bit 31: MultiCast Controller

Bit 30: SSRAM, Shared Ram

Bit 29: Arbitration

Bit 28: Serial / Deserializer

Bit 27: Data Path

Bit 26: SPI Address Space

Bit 25: Switch Mailbox RAM

Bit 24: Switch Risc Core Registers- DMA, IRQ, Switch Configuration Regs, etc.

Bus Timer: A bus timer exists in the BIF which forces a termination after a presettable amount of time. All 24 bits are used when the SPI is being addressed, only the lower 8 bits are used when any other module is being addressed. The units for the timer are in core-clock ticks. For a nominal 250MHz core clock frequency, use 4 ns per tick. The reset value is 0x40020.

13.3.6 Host to Core Mailbox Interrupts

31	28	27	24	23	20	19	16	15	12	11	8	7	0	Address Offset																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														
Host 2 Core Irq #0								Host 2 Core Irq #1								reserved								0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

When the host writes to the mailbox interrupt bits, an interrupt will be signaled to the internal Tensilica master processor. The Tensilica shall be able to clear the interrupt via writing a 1 to the offending bit. There are 2 interrupts that these bits are wired to, eight for each interrupt. In this way there will be a method to create a high and a low priority interrupt for mailbox message requests and acknowledgements.

13.3.7 Core to Host Push Mailbox Interrupts

31	28	27	24	23	20	19	16	15	12	11	8	7	0	Address Offset													
Core 2 Host Irq #0								Core 2 Host Irq #1								reserved								0			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Reset Value

The Tensilica can program any of these bits, which will generate a host interface interrupt. There are 2 groups of signals, each connected to a separate interrupt. In this way there will be a method to create a high and a low priority interrupt for mailbox requests and acknowledgements.

The bits are cleared, in general by the host interface writing to its core 2 host interrupt register.

13.3.8 SSRAM Configuration

See Risc Chapter on SSRAM

13.3.9 Interrupt Configuration, Status Registers

See Risc Chapter on Interrupt Controller

13.3.10 Watchdog Timer

See Risc Chapter on Watchdog Timer

13.3.11 UART Registers

See Uart Chapter 11

The switch chip will have 2 reset pins. One is for hardware reset, the other for software reset. The hardware reset will be used for power- up reset, power glitches and to re-initialize the entire system. Every register shall be cleared or set to its default value when the hardware reset is toggled. The software reset will actually cause an interrupt to the microprocessor, which can then go and decide which subsystems should be reset. Upon a software reset, the local processor can do some housekeeping and saving certain state information as well as decide to warm boot or cold boot.

The Tensilica can write to and reset any one of the bits in the reset register. The bits in this register need to be toggle bits... Writing a zero to them has no effect, writing a 1 to them will either set or clear it, depending on its state.

13.3.12 Tensilica Reset Register

There will exist a register which is only cleared upon a cold (hreset). This register will allow the Tensilica to know why it reset itself or was reset in the event of a watchdog timer time-out.

[illegible]**Table 13-2: Microprocessor Software Reset Register**

14 Host Interface

14.1 Overview

The host interface is a parallel external interface to the iTAP Switch chip. This interface will be used to do the following:

- Exchange messages with the iTAP Switch Tensilica Processor
- Download boot code for the Tensilica Processor
- Read and Write Configuration Registers
- Fully operate the switch in applications where the iTAP Switch Tensilica Processor is not used.

The host interface is designed to be a glueless interface with a Motorola 68360 operating in asynchronous mode with a 16 bit port size, it is, however, an asynchronous interface so any speed device could be used as long as the signals were compatible. Since there are literally hundreds of RAMs in the iTAP Switch, this interface will support a page mode to access these registers. The host interface will support a 16Kbyte memory space (13 address lines). This address space will be divided into 2 regions, a paged region which allows access to the internal switch registers and a 'fixed' address region which contains host interface configuration registers and 4 mailboxes to exchange data with the iTAP Switch Tensilica.

- Page Size is 15.5K bytes.
- Fixed Contents is 512 bytes.

Address	Memory Space
0x0000	Paged Memory
0x3E00	Fixed Contents

Address	Memory Space
0x3E00	Page Register
Other 'Fixed' Registers	
0x3100	Hi Priority Mailbox (IN)
0x2140	Low Priority Mailbox (IN)
0x2180	Hi Priority Mailbox (OUT)
0x21C0	Low Priority Mailbox (OUT)

Table 14-1: Host Interface Address Map

For normal use, it is expected that the host will communicate to the Switch via several mailboxes. These mailboxes will allow for the passing of low and high priority control messages to the local processor inside of the Switch. There is also a mode where the processor may not be used which gives the host full access to all of the iTAP Switch's registers. In fact, the host could run the switch and act as a surrogate local processor.

This document outlines the register map of the host interface, then follows with a description of the interfaces which it will need need, followed by some implementation notes.

14.2 Programmable Page Memory Map

The following table is a summary of the valid page numbers, and what they page in. For the complete list, please consult the Appendix.

Page Number	Contents
000	Undefined (mirror of page 1)
001	SPI
255	
256	
256	Data Path Link 0 CSR
257	Data Path Link 1 CSR

Proprietary and Confidential Information of Onex Communications Corporation

258	Data Path Link 2 CSR
259	Data Path Link 3 CSR
260	Data Path Link 4 CSR
261	Data Path Link 5 CSR
262	Data Path Link 6 CSR
263	Data Path Link 7 CSR
264	Data Path Link 8 CSR
265	Data Path Link 9 CSR
266	Data Path Link 10 CSR
267	Data Path Link 11 CSR
268	Data Path Global CSR
269	not currently assigned
270	not currently assigned
271	not currently assigned
272	Grant Mapper
273	Grant Demapper
274	Arbitration Stats & Regs
275	DMA
276	Misc Registers
277	MailBox Ram
278	UART
...	...
512	External Ram Start
767	External Ram End
768	Shared RAM Start

Table 14-2: Page Number Decoding

14.2.1 Paged Memory Accesses

The paged memory space will perform writes just as expected, but reads are more complex due to the nature of the data being read. Many of the iTAP switch registers will be counters, which may need to be bigger than 16 bits. To perform 32 bit reads without double buffering every counter, the following will occur:

- Reads to the Paged Memory Address space, will load 32 bits into a set of registers.
- The address +2 will also be stored.
- The 'correct' 16 bits will be output on the host interface data bus.
- If the next host access matches the stored address then the other data word will be read out on to the host interface data bus. If the next host access does not have an address match another full read is performed.

14.3 Fixed Page Memory Map

The following memory map describes the 'Fixed Contents' part of the host interface address map.

Page Number	Contents
0x3E00	Page Register
0x3E04	Interrupt Mask Register
0x3E08	Interrupt 0 Control Register

Proprietary and Confidential Information of Onex Communications Corporation

0x3E0C	Interrupt 0 Control Register
0x3E10	reserved
0x3E14	reserved
0x3E18	Programmable Interrupt Level Reg 0
0x3E1C	Programmable Interrupt Level Reg 1
0x3E20	Programmable Interrupt Level Reg 2
0x3E24	Miscellaneous Register
0x3E28	Message Status Registers
0x3E28-0x3EFC	reserved
0x3F00	Low Priority Mailbox (In)
0x3F40	Hi Priority Mailbox (In)
0x3F80	Low Priority Mailbox (Out)
0x3FC0	Hi Priority Mailbox (Out)

Table 14-3: Fixed Contents Address Map

14.3.1 Page Register

This is a 32 bit read/write register. Writes have an immediate effect of what the lower host interface address space contents are.

14.3.2 Interrupt Registers

The host interface will need access to all of the same interrupt sources as the Tensilica. It must have the ability to receive the same interrupts as the Tensilica as well as have the ability to program different interrupts at different priorities. To facilitate this, the same Interrupt Controller that the Tensilica will be using will be instantiated here. This will give the host the greatest flexibility.

The following registers are local to the Host Interface. The interrupt mapping will be the same as that given in Chapter 9, Interrupt Controller. Although each interrupt source is given 4 bits of priority encoding, only the lsb of each of the nibble fields will be used.

Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x3E04	Interrupt Mask Register																															
0x3E08	Interrupt 0 Register																															
0x3E0C	Interrupt 1 Register																															
0x3E10	reserved																															
0x3E14	reserved																															
0x3E18	PI00		PI01		PI02		PI03		PI04		PI05		PI06		PI07																	
0x3E1C	PI08		PI09		PI10		PI11		PI12		PI13		PI14		PI15																	
0x3E20	PI16		PI17		PI18		PI19		PI20		PI21		PI22		PI023																	

There will be 2 added interrupts (#22-23):

- Hi Priority Mailbox Interrupt
- Low Priority Mailbox Interrupt

14.3.3 Misc Register

The miscellaneous register controls the bus error modes for the switch.

Address	31	30	29	28	27	26	25	24
0x3E24	Misc Register							

	0	0	PMWL	PMRL	0	0	CRST	PRST
--	---	---	------	------	---	---	------	------

All of these bits are 'toggle' bits, meaning writes of 0 have no effect on them. To clear or set any of these bits, write a logical 1 to them.

PMWL - Page Memory Write Lockout

If this bit is set, any writes to the paged memory will not return a dsack. This is to prevent the host from setting bits while in message-passing mode. In fabrics that do not use the iTAP Switch local microprocessor, this bit would be cleared. This bit powers up to a zero.

PMRL - Page Memory Read Lockout

If this bit is set, any reads to the paged memory will not return a dsack. This is to prevent the host from getting any status information while in message-passing mode. During debug, even in message passing mode this bit will generally be cleared to ensure that random diagnostics reads do not bus error the system. However, application software may want to set this bit to debug a system where a bad pointer may be reading an errant location.

The reset register gives the host the ability to reset the iTAP switch chip, or the Switch chip microprocessor sub-system.

CRST - Chip Reset

Writing a 1 to this register will reset the entire chip, including the host interface when the access is complete.

PRST - Processor Reset

Writing a 1 to this register will place the Switch microprocessor into reset. To take the processor out of reset the host interface will need to toggle this bit to zero by writing a 1 to it. The power up status of this register will be the value on the BootMode[1] pin (which is used to enable or disable initial booting of the internal Master Processor).

14.3.3.1 Host Interface BootStrap Mechanism

When the iTAP Switch Processor is in reset (PRST is set) code may be safely downloaded to the boot ram. This is done by setting the page register to 'BootCode' and writing the boot image in. The image may be read back at any time to ensure that it was delivered successfully. When the host has correctly put the boot image into the BootCode space, it will then clear the PRST bit. At this time, the local switch processor will begin executing code from its reset vector.

14.3.4 Message Status Registers

The control structures for these mailboxes are located below in the fixed contents area:

Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x3E28	High Priority Message In				High Priority Message Out				Low Priority Message In				Low Priority Message Out																			
	Host to Master	Master to Host	Host to Master	Master to Host	Host to Master	Master to Host	Host to Master	Master to Host	Host to Master	Master to Host	Host to Master	Master to Host	Host to Master	Master to Host	Host to Master	Master to Host	Host to Master	Master to Host	Host to Master	Master to Host	Host to Master	Master to Host	Host to Master	Master to Host	Host to Master	Master to Host	Host to Master	Master to Host	Host to Master	Master to Host	Host to Master	Master to Host

This register is divided into 4 sections, one for each mailbox- High Priority Message In, Message Out, and Low Priority Message In and Message Out. Each one of the mailbox status registers has 4 bits which the host can set that causes an interrupt on the master processor. There are also 4 bits which the master processor can set which cause an interrupt on the host processor. The 2 High Priority Mailboxes are on interrupt #X and the 2 Low Priority Mailboxes are on interrupt #Y. It is expected that the host and master processor will program the high priority mailboxes on a different (and higher) interrupt than the low priority mailboxes to facilitate 2 classes of messages. To clear any of these bits, the host processor must write a logical 1 to them. This will ensure that if during the processor of clearing a bit the master processor attempts to set another bit, it is not overwritten. The hardware makes no restrictions or assumptions other than connecting one interrupt to the High Priority Response bits, and other to the Low Priority Response bits. The bits are aligned such that the 16 bit host interface can set and clear a mailbox status register in a single write cycle- i.e. clearing the interrupt condition as well as giving its response if software desires.

Please consult the software documentation concerning the usage of these bits.

Proprietary and Confidential Information of Onex Communications Corporation

14.3.5 Host Interface Mailboxes

For messages in and out of the switch there are high and low priority mailboxes. The format of these mailboxes is as follows:

Address	
0x3F00	In Mailbox, Low Priority- 64 bytes
0x3F04	
0x3F08	
0x3F3C	
0x3F40	In Mailbox, High Priority- 64 bytes
0x3F80	Out Mailbox, Low Priority- 64 bytes
0x3FC0	Out Mailbox, High Priority- 64 bytes

14.4 Interfaces

The host interface has 3 ports- the 68360 external bus, the mailbox bus and the HIF interface. Each are described below. In the block diagram the address, data and byte enable control signals have been omitted for clarity.

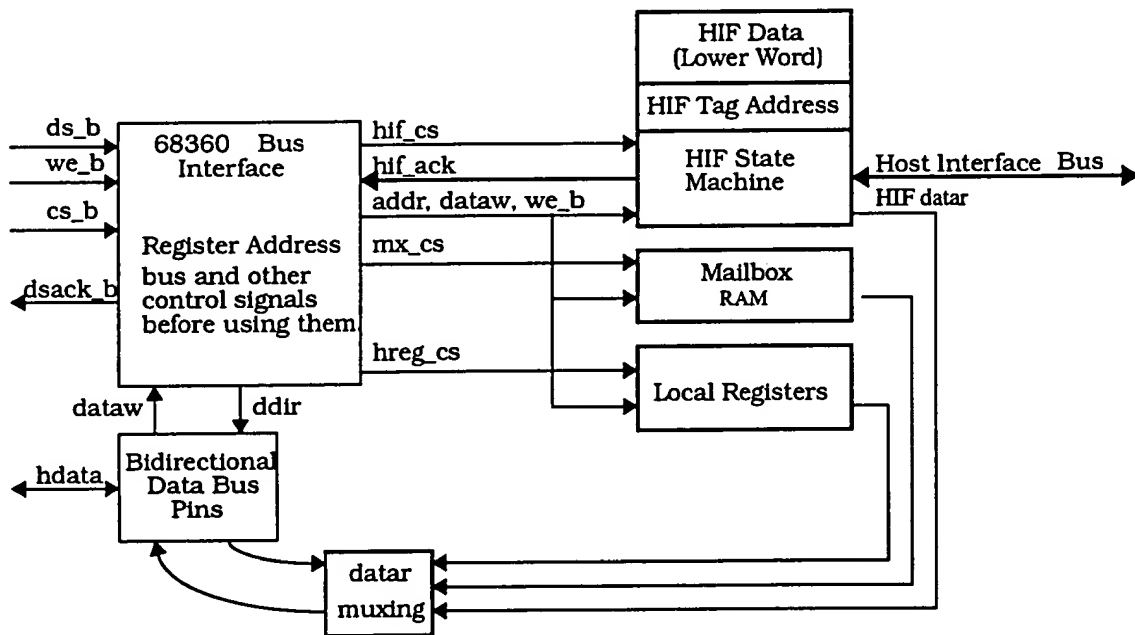


Figure 14-4: Host Interface Block Diagram

14.4.1 Motorola 68360 Bus Interface

The host bus interface will have the following pins:

Pin Name	Width	I/O	Description
hcs_b	1	I	Chip Select (Active low)
hds_b	1	I	Data Strobe(Active low)
hwe_b[1:0]	2	I	Read=1, Write=0 hwe_b[1] -> dbus bits[31:24] hwe_b[0] -> dbus bits[23:16]

haddr	14	I	Address Pins
hdata	16	B	Data Bus
irq0_b	1	O	Interrupt 0
irq1_b	1	O	Interrupt 1
hdsack_b	1	O	Transfer Ack (Active low)-opendrain

The 68360 needs to be programmed to output individual byte write-enables, as well as accept external data transfer acknowledges. Since the 68360 has a 32 bit data bus, the bus needs to be connected as follows:

68360 Pin	ITAP Switch Pin
addr[13:1]	haddr[12:0]
data[31:16]	hdata[15:0]
ds_b	hds_b
we_b[1]	hwe_b[1]
we_b[0]	hwe_b[0]
dsack_b[1]	hdsack_b
dsack_b[0]	-
irq[x]_b	irq0_b
irq[y]_b	irq1_b

This will always signal a 16 bit port size to the 68360. Refer to 68360 User's Manual Table 4-2 'DSACKx Encoding'. The 68360's dsack0 pin should be pulled up.

14.4.1.1 68360 Device Settings

- The following registers will need to be set in it:
PEPAR -> Bit 7, set to 1. Set this to '1' so that WE_b[3:0] are driven. Out of reset, the 68360 tri-states these pins, so they will need pull-up resistors.
SPS[1:0]-> In the Option Register, need to be set to '11' for external dsack generation.

14.4.1.2 Timing Diagrams

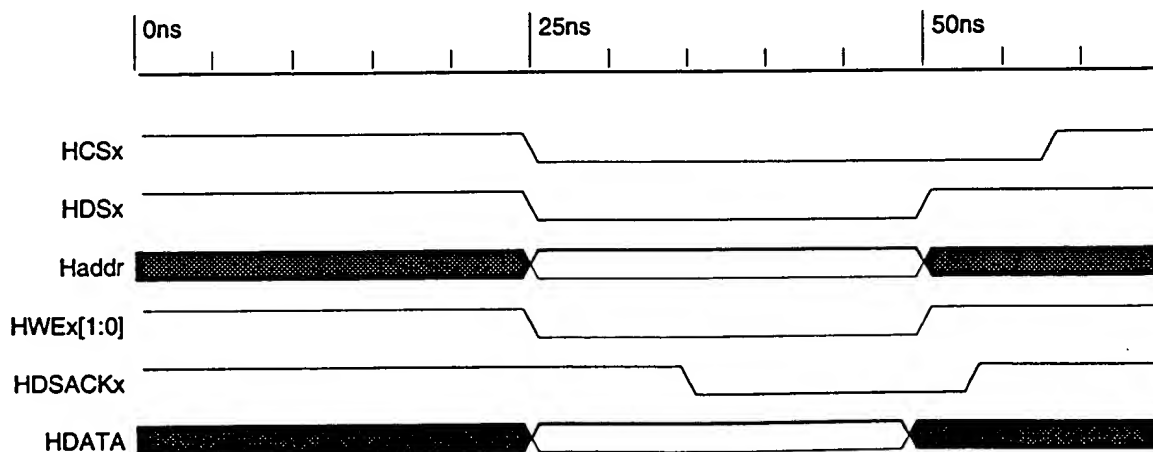


Figure 14-5: Host Interface Write Timing Diagram

Proprietary and Confidential Information of Onex Communications Corporation

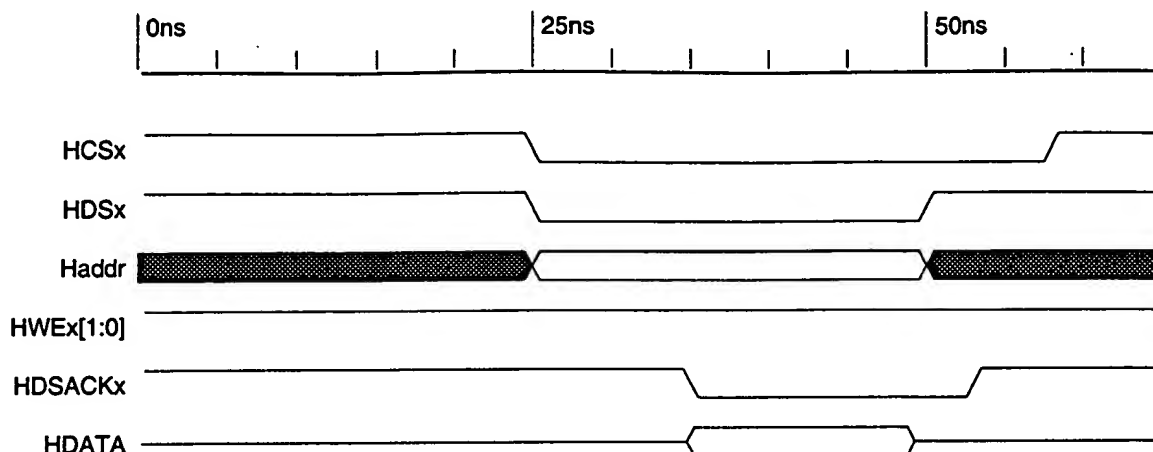


Figure 14-6: Host Interface Read Timing Diagram

Note: Since this is an asynchronous interface, the absolute timing is unimportant.

14.4.1.3 Bus Calculations

The time the 68360 spends waiting on the host interface can be calculated as follows:

Worst Case Interface:

- Local Registers- 0 ws
- Mailbox Ram- 0 ws
- SBus Interface- 2 interfaces w/ longword xfers ahead of host interface
- FBus Interface- 1 lw, 1 line xfer ahead of the host interface

Accesses to the FBus need to take into account the following delays:

- Bring DSx into ITAP Switch Clock Domain (2clks)
- Sample, assert HIF Request (2 clks)
- Arbitrate for FBus.
 - DMA Line Access to external SSRAM = 13clks
 - Processor LW Access to external SSRAM = 8clks
 - Host IF LW Access to external SSRAM = 8 clks
- Handskaking to Dsack assertion = 3clks

Total latency is 36 internal 250MHz clock cycles. This will incur ~ 150ns of latency on the bus, which at 50MHz is 8 clock cycles and at 33MHz is 5 clock cycles.

The 2 typical cases are when the Tensilica is operating and the host interface is using the mailboxes(which has 0 ws) and when the Tensilica is not operating and the host interface is controlling the switch via the SBus. Sbus latency will be 2 long word acceses. Assuming the Sbus runs at 1 ws, the latency from DSx to DSackx will be 15 core clock cycles. At 250MHz this 60ns, and translates to 2/3 (33/50 MHz) host interface wait states to write a 16 bit word. For reads to the 32 bit internal registers, 1 read cycle with the wait states will pre-load the second part of the word, so that it may be accessed with 0 wait states.

14.4.2 HIF Bus Interface

The HIF bus will have the capability of performing 32 bit read/write transfers to both the SBus and the FBus. It will not support any of the FBus bursting modes. This is because there is little buffering in the Host Interface and the 68360 bus interface can't keep up. The HIF will allow the external host to control the iTAP Switch chip just as the embedded Tensilica processor can. All of the

Proprietary & Confidential Information of Onex Communications Corporation

same memory mapped registers and interrupts are available to it. This state machine will take the transactions from the 68360 bus and perform a bus conversion to a simplified Tensilica PIF bus. This bus will not support block reads since more buffering would be needed, and the expected interface speed (PCI translated to 68360 async I/O) does not have the bandwidth adequate to support block reads at 250MHz.

14.4.3 Mailbox Ram Bus Interface

The Mailbox ram interface gives the host fast access to a small dual port ram which it can communicate to the local Tensilica processor via a messaging scheme which is TBD. This is a single cycle RAM so there are no latency or acknowledge signals which need to be passed back to the controller. It will merely pass along the address, data and byte enable control signals. Since the 68360 does not support unaligned transfers the the host interface's 16 bit data bus will be replicated as it goes into the Mailbox Ram.

This interface tightly couples onto an SRAM, there will be byte writes. The size of this ram will be a true dual port RAM that is 64 entries x 32bits wide.

14.5 Host Interface Top Level I/O

Pin Name	Width	I/O	Description
<i>Host Interface Pins</i>			
hcs_b	1	I	Chip Select (Active low)
hds_b	1	I	Data Strobe(Active low)
hwe_b[1:0]	2	I	Read=1, Write=0
haddr[12:0]	13	I	Address Pins
hdata	16	B	Data Bus
irq0_b	1	O	Interrupt 0
irq1_b	1	O	Interrupt 1
hdsack_b	1	O	Transfer Ack (Active low)-opendrain
<i>Mailbox RAM Interface</i>			
maddr	6	O	Mailbox address-longword aligned
mdataw	32	O	Mailbox Write data bus
mdatar	32	I	Mailbox Read data bus
mwe	4	O	Mailbox Write Enable
mcs	1	O	Mailbox CS
<i>HIF interface</i>			
HIFCnt	1	O	HIF Control Bits
HIFValid	1	O	HIF Valid
BIFCntI	2	I	BIF Module Control Bits
BIFValid	1	I	BIF Valid
BIFReqRdy	1	I	BIF Module Ready
HIFAddr	x	O	HIF Address
HIFdataW	32	O	HIF Write Data bus
HIFdataR	32	I	HIF Read Data bus
HIFBe	4	O	HIF Byte Enables
<i>Misc</i>			
sysclock	1	I	System Clock
reset_b	1	I	Reset (Active Low)
pirq	8	I	Processor Interrupts

sleep	1	1	Processor Sleep
-------	---	---	-----------------

14.6 Notes

1. 68360 can abort the bus cycle, make sure we can handle this correctly.
2. Double buffer the control inputs, they're asynchronous.
3. The 68360 16 bit port size will cause the following to be decoding:

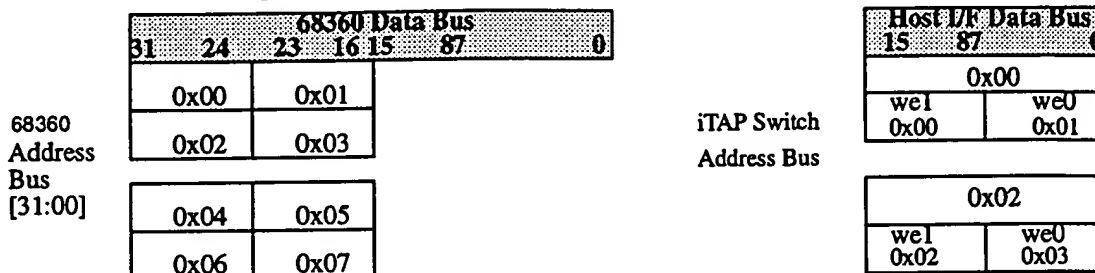


Figure 14-7: Big Endian 16 bit 68360 Port Addressing

4. The size of the host interface will be
 - 64x32bit Dual Port Ram
 - 32 bit read buffers
 - 13 bit address tag buffer
 - Local Registers- 10 page bits, Irq~80, 10 misc, 25 for FSMs and output registering
 - Metastability - 34 FFs, (need to sample data bus at the correct time, so we don't need to register it).

Total Flip Flops~ 210 + Dual Port RAM of 2048 bits.

Proprietary and Confidential Information of Onex Communications Corporation

15 Configuration/Status Register List

This section summarizes the Control Status Registers (CSRs) used to configure and monitor the operation of the ITSE. CSRs include all configurable memory devices within the ITSE, these devices may be individual flip-flops, register arrays or memory arrays.

Table 15-1: Datapath CSRs

31	24	23	16	15	8	7	0	Address Offset
								0x0000
								0x0004

15.1 Memory Map:

The memory map of the chip is 32 megabytes. All registers are accessible via the host interface by setting the page register accordingly.

Bus Mapping	Block Description	Address Range	Host Interface Page Range	Size (MB)
SBus	SPI Memory Space	0x0000 0000	0x000	4
		0x003F FFFF	0x0FF	
SBus	DataPath CSRs	0x0040 0000	0x100	4
		0x0043 0000	0x10C	
SBus	Synchronizers	0x0043 4000	0x10D	
SBus	Arbitration	0x0044 0000	0x110	
		0x0044 8000	0x112	
SBus	DMA	0x0044 C000	0x113	
SBus	Misc Registers	0x0045 0000	0x114	
SBus	MailBox Ram	0x0045 4000	0x115	
SBus	UART	0x0045 8000	0x116	
FBus	External RAM	0x0080 0000	0x200	4
		0x007F FFFF	0x2FF	
FBus	Internal Shared Ram	0x00C0 0000	0x300	0.032
		0x00C04000	0x301	

15.1.1 Synchronizer Memory Map

Base Address 0x00434000

31	24	23	16	15	8	7	0	Address Offset
0	0	0	frame_size				0	0x0000
0	0	0	lostsync_loop_limit	0	0	0	0	0x0004
0	0	0	presync_loop_limit	0	0	0	0	0x0008
0	0	0	gate_capture_data_position				0	0x000C
0	0	0	total_data_slot_limit				0	0x0010
0	0	0	data_row_toggle_position				0	0x0014
0	0	0	total_grant_slot_limit				0	0x0018
0	0	0	grant_row_toggle_position				0	0x001C
0	0	0	data_framing_pattern_position				0	0x0020
0	0	0	data_sor_offset				0	0x0024
0	0	0	data_oslot_num_start_position				0	0x0028
0	0	0	data_link_input_enable				0	0x002C
0	0	0	data_link_output_enable				0	0x0030
0	0	0	start_oslot_position				0	0x0034
0	0	0	sync_changed_dc1_int_mask				0	0x0038
0	0	0	sync_changed_gc_int_mask				0	0x003C
0	0	0	sync_error_limit_dc1_int_mask				0	0x0040
0	0	0	sync_error_limit_gc_int_mask				0	0x0044
0	0	0	crc_error_limit_dc1_int_mask				0	0x0048
0	0	0	crc_error_limit_gc_int_mask				0	0x004C
0	0	0	data_slot_sync_error_int_mask				0	0x0050
0	0	0	sync_status_dc1				0	0x0054
0	0	0	sync_status_gc				0	0x0058
0	0	0	data_slot_sync_error				0	0x005C
0	0	0	sync_error_limit_dc1				0	0x0060
0	0	0	sync_error_limit_gc				0	0x0064
0	0	0	crc_error_limit_dc1				0	0x0068
0	0	0	crc_error_limit_gc				0	0x006C
0	0	0	pp_sync_offset_count_l0				0	0x0070
0	0	0	pp_sync_offset_count_l2				0	0x0074
0	0	0	pp_sync_offset_count_l4				0	0x0078
0	0	0	pp_sync_offset_count_l6				0	0x007C

Base Address 0x00434000

Year	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100
1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	

Proprietary and Confidential Information of Onex Communications Corporation

Base Address 0x00434000

31	24	23	16	15	8	7	0	Address Offset
0	0	0	0	0	0	0	0	lost_sync_error_count_gc_l6 0x0178
0	0	0	0	0	0	0	0	lost_sync_error_count_gc_l7 0x017c
0	0	0	0	0	0	0	0	lost_sync_error_count_gc_l8 0x0180
0	0	0	0	0	0	0	0	lost_sync_error_count_gc_l9 0x0184
0	0	0	0	0	0	0	0	lost_sync_error_count_gc_l10 0x0188
0	0	0	0	0	0	0	0	lost_sync_error_count_gc_l11 0x018c
0	0	0	0	0	0	0	0	crc_error_count_dc0_l0 0x0190
0	0	0	0	0	0	0	0	crc_error_count_dc0_l1 0x0194
0	0	0	0	0	0	0	0	crc_error_count_dc0_l2 0x0198
0	0	0	0	0	0	0	0	crc_error_count_dc0_l3 0x019c
0	0	0	0	0	0	0	0	crc_error_count_dc0_l4 0x01a0
0	0	0	0	0	0	0	0	crc_error_count_dc0_l5 0x01a4
0	0	0	0	0	0	0	0	crc_error_count_dc0_l6 0x01a8
0	0	0	0	0	0	0	0	crc_error_count_dc0_l7 0x01ac
0	0	0	0	0	0	0	0	crc_error_count_dc0_l8 0x01b0
0	0	0	0	0	0	0	0	crc_error_count_dc0_l9 0x01b4
0	0	0	0	0	0	0	0	crc_error_count_dc0_l10 0x01b8
0	0	0	0	0	0	0	0	crc_error_count_dc0_l11 0x01bc
0	0	0	0	0	0	0	0	crc_error_count_dc1_l0 0x01c0
0	0	0	0	0	0	0	0	crc_error_count_dc1_l1 0x01c4
0	0	0	0	0	0	0	0	crc_error_count_dc1_l2 0x01c8
0	0	0	0	0	0	0	0	crc_error_count_dc1_l3 0x01cc
0	0	0	0	0	0	0	0	crc_error_count_dc1_l4 0x01d0
0	0	0	0	0	0	0	0	crc_error_count_dc1_l5 0x01d4
0	0	0	0	0	0	0	0	crc_error_count_dc1_l6 0x01d8
0	0	0	0	0	0	0	0	crc_error_count_dc1_l7 0x01dc
0	0	0	0	0	0	0	0	crc_error_count_dc1_l8 0x01e0
0	0	0	0	0	0	0	0	crc_error_count_dc1_l9 0x01e4
0	0	0	0	0	0	0	0	crc_error_count_dc1_l10 0x01e8
0	0	0	0	0	0	0	0	crc_error_count_dc1_l11 0x01ec
0	0	0	0	0	0	0	0	crc_error_count_dc0_l0 0x01f0
0	0	0	0	0	0	0	0	crc_error_count_gc_l1 0x01f4
0	0	0	0	0	0	0	0	crc_error_count_gc_l2 0x01f8
0	0	0	0	0	0	0	0	crc_error_count_gc_l3 0x01fc

Proprietary and Confidential Information of Onex Communications Corporation

Base Address 0x00434000

31	24	23	16	15	8	7	0	Address Offset
0	0	0	0	0	0	0	0	crc_error_count_gc_l4 0x0200
0	0	0	0	0	0	0	0	crc_error_count_gc_l5 0x0204
0	0	0	0	0	0	0	0	crc_error_count_gc_l6 0x0208
0	0	0	0	0	0	0	0	crc_error_count_gc_l7 0x020c
0	0	0	0	0	0	0	0	crc_error_count_gc_l8 0x0210
0	0	0	0	0	0	0	0	crc_error_count_gc_l9 0x0214
0	0	0	0	0	0	0	0	crc_error_count_gc_l10 0x0218
0	0	0	0	0	0	0	0	crc_error_count_gc_l11 0x021c

The Address Offset range 0x0300 - 0x041c has the same read values as the range 0x0100 - 0x021c except that upon a read, the register is cleared. A write to the range 0x0300 - 0x041c is not allowed and will result in a bus error.

Table 15-2: Synchronizer Misc Contol Bit Fields

Bit#	Field
0	disable_scrambler
1	disable_descrambler
2	bypass_sync
3	switch_sor_reset
5:4	slot_size_select

15.1.2 Arbitration Memory Map

Base Address 0x00440000

31	24	23	16	15	8	7	0	Address Offset
Grant Mapper Ram Slot 0								0x0000
								0x0004
Grant Mapper Ram Slot 1								0x0008
								0x000C
Grant Mapper Ram Slot 849								0x1A88
								0x1A8C
unmapped address space								0x1A90
Grant DeMapper Ram Slot 0								0x4000
								0x4004

Proprietary and Confidential Information of Onex Communications Corporation

Base Address 0x00440000

31	24	23	16	15	8	7	0	Address Offset
Grant DeMapper Ram Slot 1								0x4008
								0x400C
Grant DeMapper Ram Slot 849								0x5A88
								0x5A8C
unmapped address space								0x5A90
Grant Link 00 Status Reg								0x8000
Grant Link 01 Status Reg								0x8004
Grant Link 02 Status Reg								0x8008
Grant Link 03 Status Reg								0x800C
Grant Link 04 Status Reg								0x8010
Grant Link 05 Status Reg								0x8014
Grant Link 06 Status Reg								0x8018
Grant Link 07 Status Reg								0x801C
Grant Link 08 Status Reg								0x8020
Grant Link 09 Status Reg								0x8024
Grant Link 10 Status Reg								0x8028
Grant Link 11 Status Reg								0x802C
Grant Link Framing Pattern								0x8030
	Grant Link Framing Pattern	read only- all zeroes						0x8034
Grant Link Common 'Stuff' Reg								0x8038
Grant Link 00 Line Overhead Status Reg								0x803C
Grant Link 01 Line Overhead Status Reg								0x8040
Grant Link 02 Line Overhead Status Reg								0x8044
Grant Link 03 Line Overhead Status Reg								0x8048
Grant Link 04 Line Overhead Status Reg								0x804C
Grant Link 05 Line Overhead Status Reg								0x8050
Grant Link 06 Line Overhead Status Reg								0x8054
Grant Link 07 Line Overhead Status Reg								0x8058
Grant Link 08 Line Overhead Status Reg								0x805C
Grant Link 09 Line Overhead Status Reg								0x8060
Grant Link 10 Line Overhead Status Reg								0x8064
Grant Link 11 Line Overhead Status Reg								0x8068

Proprietary Confidential Information of Onex Communications Corporation

Base Address 0x00440000

31	24	23	16	15	8	7	0	Address Offset
Grant Max Grants								0x806C
								0x8070
								0x8074
Grant Capture Interrupt Status Register								0x8078
Grant Link 0 Capture Register Contents								0x807C
Grant Link 1 Capture Register Contents								0x8080
Grant Link 2 Capture Register Contents								0x8084
Grant Link 3 Capture Register Contents								0x8088
Grant Link 4 Capture Register Contents								0x808C
Grant Link 5 Capture Register Contents								0x8090
Grant Link 6 Capture Register Contents								0x8094
Grant Link 7 Capture Register Contents								0x8098
Grant Link 8 Capture Register Contents								0x809C
Grant Link 9 Capture Register Contents								0x80A0
Grant Link 10 Capture Register Contents								0x80A4
Grant Link 11 Capture Register Contents								0x80A8
Grant Link 0 Capture Register Bit Mask								0x80AC
Grant Link 1 Capture Register Bit Mask								0x80B0
Grant Link 2 Capture Register Bit Mask								0x80B4
Grant Link 3 Capture Register Bit Mask								0x80B8
Grant Link 4 Capture Register Bit Mask								0x80BC
Grant Link 5 Capture Register Bit Mask								0x80C0
Grant Link 6 Capture Register Bit Mask								0x80C4
Grant Link 7 Capture Register Bit Mask								0x80C8
Grant Link 8 Capture Register Bit Mask								0x80CC
Grant Link 9 Capture Register Bit Mask								0x80D0
Grant Link 10 Capture Register Bit Mask								0x80D4
Grant Link 11 Capture Register Bit Mask								0x80D8
Gnt Config		0	0	0	0	0	0	Grant Parity Error Masks
Grant Error Interrupt Mask								0x80DC
Grant Error Interrupt Status Register								0x80E4
								Grant Parity Error
								0x80E8
								Grant DeMapper Sequence Error
								0x80EC

Proprietary Confidential Information of Onex Communications Corporation

Base Address 0x00440000

31	24 23												16 15												8 7												0												Address Offset																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
Grant Dest Error Element[31:00]																																																0x80F0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
Grant Dest Error Element[47:32]																								reserved																								0x80F4																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
unmapped memory space																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
Request Link 0 Priority 0 Statistics																																																0x9000																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
Request Link 0 Priority 1 Statistics																																																0x9004																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
Request Link 0 Priority 2 Statistics																																																0x9008																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
Request Link 0 Priority 3 Statistics																																																0x900C																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
Request Link 0 Priority 4 Statistics																																																0x9010																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
Request Link 0 Priority 5 Statistics																																																0x9014																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
Request Link 0 Priority 6 Statistics																																																0x9018																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
Request Link 0 Priority 7 Statistics																																																0x901C																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
Request Link 1 Priority 0 Statistics																																																0x9020																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
Request Link 2 Priority 0 Statistics																																																0x9040																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
Request Link 3 Priority 0 Statistics																																																0x9060																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
Request Link 4 Priority 0 Statistics																																																0x9080																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
Request Link 5 Priority 0 Statistics																																																0x90A0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
Request Link 6 Priority 0 Statistics																																																0x90C0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
Request Link 7 Priority 0 Statistics																																																0x90E0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
Request Link 8 Priority 0 Statistics																																																0x9100																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
Request Link 9 Priority 0 Statistics																																																0x9120																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
Request Link 10 Priority 0 Statistics																																																0x9140																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
Request Link 11 Priority 0 Statistics																																																0x9160																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
0	Max Request Link 00			0	Num Requests Link 00			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Proprietary and Confidential Information of Onex Communications Corporation

15.1.2.1 Grant Mapper Ram

31	28	27	24	23	20	19	16	15	12	11	8	7	0	Address Offset
Grant Link 00	Grant Link 01	Grant Link 02	Grant Link 03	Grant Link 04	Grant Link 05	0	0	0	0	0	0	0	0	0x0000
Grant Link 06	Grant Link 07	Grant Link 08	Grant Link 09	Grant Link 10	Grant Link 11	0	0	0	0	0	0	0	0	0x0004

Bits 31-8 are read/writable, bits 7-0 read back zero always. Upon reset the registers will contain random patterns and must be written to.

Bit Coding is:

- 0000 - Idle
- 0001 - GE0
- 0010 - GE2
- 0011 - GE3
- 1000 - LOH Framing Pattern
- 1001 - LOH Status
- 1010 - LOH ID
- 1011 - LOH Stuff
- 1100 - LOH Sync

15.1.2.2 Grant DeMapper Ram

31	30	28	26	24	23	22	20	18	16	15	14	12	10	8	7						0	Address Offset
0	Grant Link 00	0	Grant Link 01	0	Grant Link 02	0	Grant Link 03	0	Grant Link 04	0	Grant Link 05	0	0	0	0	0	0	0	0	0	0x4000	
0	Grant Link 06	0	Grant Link 07	0	Grant Link 08	0	Grant Link 09	0	Grant Link 10	0	Grant Link 11	0	0	0	0	0	0	0	0	0	0x4004	

Reset value is random, they must be programmed before use.

Bit Coding is:

- 000 - Idle
- 001 - GE0
- 010 - GE2
- 011 - GE3
- 100 - LOH Capture

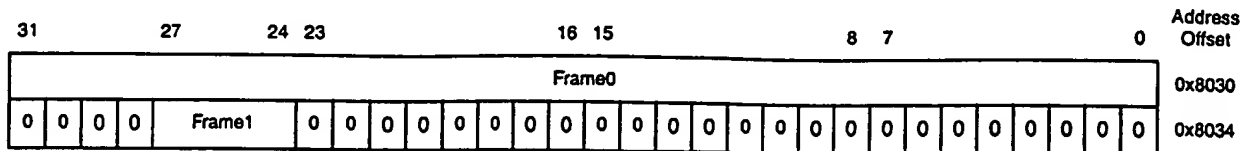
15.1.2.3 Grant Link Status Register

31	29	24	23	22	16	15	14	8	7	6	0	Address Offset
fifo full	0	Current Fifo Watermark	0	Grants Received Last Row	0	Grants Dropped Last Row	0	Grants Forwarded Last Row				0x8000-0x802C

This is a read only register, updated at the end of every row.

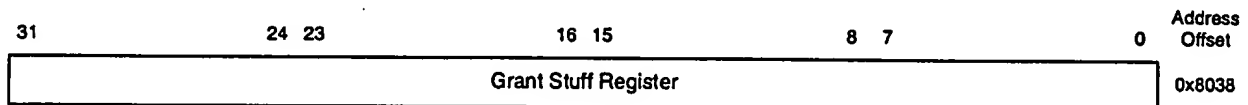
Proprietary and Confidential Information of Onex Communications Corporation

15.1.3 Grant Link Framing Pattern



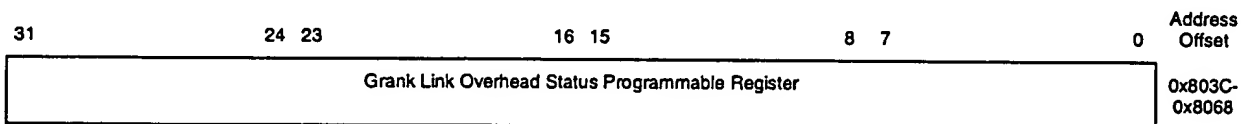
The grant link framing pattern is 36 bits wide and is made by concatenating frame0 with frame1 such that: FramingPattern[35:0] = Frame1[27:24], Frame0[31:0].. The reset value of Frame 0 is 0x0000F628 and Frame 1 is 0. So that the framing pattern is 0x0F628.

15.1.3.1 Grant Link Stuff Register



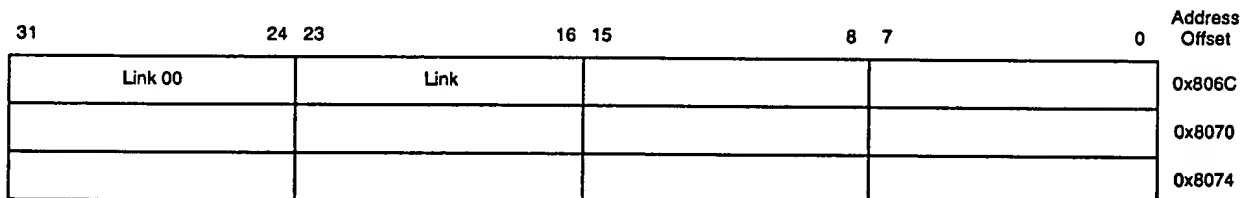
This is a 32 bit read/write register. Reset value is all zeros.

15.1.3.2 Grant Link Overhead Status Register



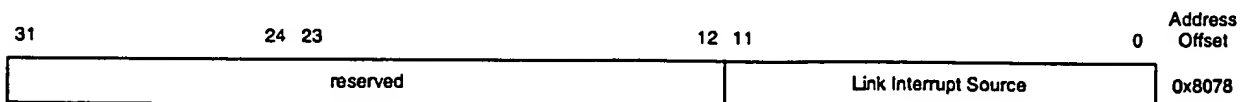
This is a 32 bit read/write register. All bits are writeable. The reset value is 0. The use of this register is still undefined.

15.1.3.3 Maximum Grants Register



The reset value of these registers allows grants up to 96 PDUs for the following row. Writes to this register are effective immediately.

15.1.3.4 Grant Capture Interrupt Status Register



Link Interrupt Source - bit is active hi if contents of capture slot are different than previous rows.

Bit 0: Link 00

Bit 1: Link 01

Bit 2: Link 02

...

Bit 11: Link 11

15.1.3.5 Grant Parity Error Masks/ Grant Config

31	24 23	12 11	0	Address Offset
Grant Config	reserved	Grant Parity Error Masks		0x80DC

Grant Parity Error Masks - bit is active hi to enable parity conformance. Reset value is 0.

Bit 0: Link 00

Bit 1: Link 01

Bit 2: Link 02

...

Bit 11: Link 11

Grant Config

Bit 7: Grant Rotate Enable: Reset Value is 0, Set to a 1 to enable grant parser rotation.

Bit 6: Disable Num Field (grants are forced to single PDU reservation mode). Default is 0.

15.1.3.6 Grant Error Interrupt Mask

31	24 23	12 11	0	Address Offset
Grant Error Interrupt Mask				0x80E0

Bit 31: Grant Start Signals Unaligned Error Mask: Reset is 0, Program to 1 to enable this type of error.

Bit 30: Grant Minimum Start Pulse Error Mask: Reset is 0, Program to 1 to enable this type of error. This error mask is ineffective for grants since there is no minimum pulse period (it is a holdover from the request parser).

Bit 29: Grant Remaining Mask: At the end of a row, if grants are remaining in the buffers, this will trigger an interrupt when set to a 1, reset value is 0.

Bit 28: Grant Fifo Filled Mask: If the grant fifo overflows for a link this will trigger when programmed to a 1, reset value is 0. Fifo watermarks need to be investigated for the link.

Bits 23-12: Grant Mapper Sequencing Mask: Set to a 1 to allow sequencing errors to generate an interrupt.

Bits 11-0: Grant Demapper Sequencing Mask: Set to a 1 to allow sequencing errors to generate an interrupt.

15.1.3.7 Grant Error Interrupt Status Register

31	24 23	12 11	7	0	Address Offset
Grant Error Interrupt Status Register	reserved				0x80E4

Bit 31: `gnt_dest_error_int` - signals that a grant element has attempted to goto an output it isn't supposed to. Write a 1 to clear this type of interrupt source.

Bit 30: `gnt_remaining_int` - signals grants are still in the fifo at the end of the row. Write a 1 to clear this interrupt source.

Bit 29: gnt_fifo_fill_int - signals that a grant fifo has overfilled. Write a 1 to clear this interrupt source.

Bit 28: gnt_start_min_error_int - signals that grant elements have arrived too quickly. Write a 1 to clear this interrupt source.

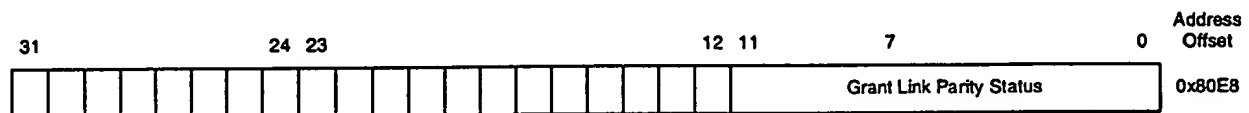
Bit 27: gnt_start_align_error_int - signals alignment error, write a 1 to clear this interrupt source

Bit 26: gnt_mapper_int - signals to read grant sequencing error register

Bit 25: gnt_demapper_int - signals to read grant sequencing error register

Bit 24: gnt_parity_int - signals to read the grant parity error register

15.1.3.8 Grant Parity Error



Bit 11: Link 11

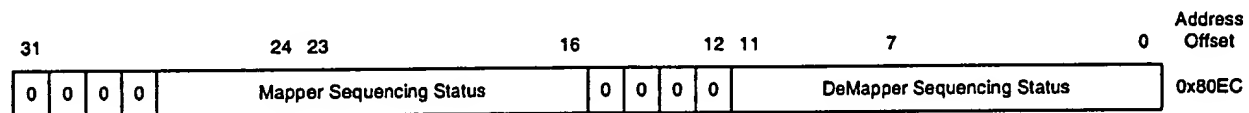
Bit 10: Link 10

....

Bit 0: Link 0

When a grant parity error is detected, writing a 1 to the bit in this register which is causing the interrupt, will clear the interrupt.

15.1.3.9 Grant Sequence Error



When a Sequencing Error is detected, this register should be read to determine the input or output link which has generated the error. Write a 1 to the offending bit to clear the interrupt.

Bit 24: Output Link 11

....

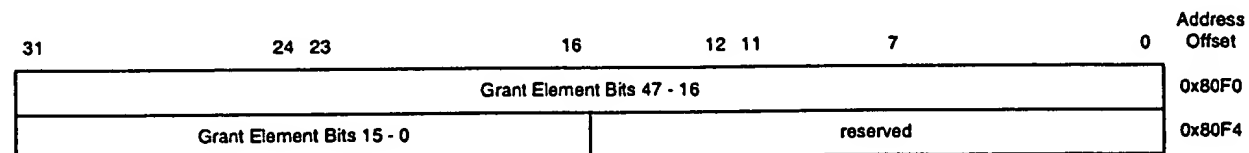
Bit 16: Output Link 0

Bit 11: Input Link 11

....

Bit 0: Input Link 0

15.1.3.10 Grant Destination Error Element



When a grant link wiring error is detected, the grant element is captured into a register and is made accessible here. This is a read only register.

Proprietary and Confidential Information of Onex Communications Corporation

15.1.3.11 Link X Request Element Priority Statistics Registers

31	27	26	23	16	15	11	10	7	0	Address Offset
0	Link 00 Priority 0 Requests Dropped				0	Link 00 Priority 0 Request Received				0x9000
0	Link 00 Priority 1 Requests Dropped				0	Link 00 Priority 1 Request Received				0x9004
0	Link 00 Priority 2 Requests Dropped				0	Link 00 Priority 2 Request Received				0x9008
0	Link 00 Priority 3 Requests Dropped				0	Link 00 Priority 3 Request Received				0x900C
0	Link 00 Priority 4 Requests Dropped				0	Link 00 Priority 4 Request Received				0x9010
0	Link 00 Priority 5 Requests Dropped				0	Link 00 Priority 5 Request Received				0x9014
0	Link 00 Priority 6 Requests Dropped				0	Link 00 Priority 6 Request Received				0x9018
0	Link 00 Priority 7 Requests Dropped				0	Link 00 Priority 7 Request Received				0x901C

This pattern repeats for each of the 12 links, the offset is 0x20 between different link's statistics registers. These registers are read - only.

15.1.3.12 Link X Request Element Counters

31		27 26		24 23		16 15		7		0		Address Offset									
0	Link 00 Maximum Requests / Row			0	Link 00 Num Requests Forwarded			0	0	0	0	0	0	0	0	0	0	0	0	0	0x9180

The Maximum Requests per row field has a reset value of 96 and is read-writeable. The number of requests forwarded refers to the previous row and is read-only.

15.1.4 Miscellaneous Registers

Base Address: 0x450000

31	24 23				16 15				8 7				0	Address Offset				
0	0	0	0	ilink_mask00				ilink_spare00				0	0	0	0	0	stage num00	0x0000
0	0	0	0	ilink_mask01				ilink_spare01				0	0	0	0	0	stage num01	0x0004
0	0	0	0	ilink_mask02				ilink_spare02				0	0	0	0	0	stage num02	0x0008
0	0	0	0	ilink_mask03				ilink_spare03				0	0	0	0	0	stage num03	0x000C
0	0	0	0	ilink_mask04				ilink_spare04				0	0	0	0	0	stage num04	0x0010
0	0	0	0	ilink_mask05				ilink_spare05				0	0	0	0	0	stage num05	0x0014
0	0	0	0	ilink_mask06				ilink_spare06				0	0	0	0	0	stage num06	0x0018
0	0	0	0	ilink_mask07				ilink_spare07				0	0	0	0	0	stage num07	0x001C
0	0	0	0	ilink_mask08				ilink_spare08				0	0	0	0	0	stage num08	0x0020

Proprietary Confidential Information of Onex Communications Corporation

Base Address: 0x450000

31					24	23					16	15					8	7					0					Address Offset
0	0	0	0	ilink_mask09				ilink_spare09				0	0	0	0	0	0	stage num09				0x0024						
0	0	0	0	ilink_mask10				ilink_spare10				0	0	0	0	0	0	stage num10				0x0028						
0	0	0	0	ilink_mask11				ilink_spare11				0	0	0	0	0	0	stage num11				0x002C						
Programmable Switch ID Number (r/w)																								0x0030				
ITAP Switch Part Number (read only)																Switch Revision (read only)								0x0034				
ITAP Switch Module Reset Register				SPI Page Register Size				spi_a	spi_sel	SPI_MODE	0	0	0	0	BootMode				0	0	CLK_SEL	WRST	0x0038					
SBus Module Full Decode				SBus Timer																				0x003C				
Host to Core Push Button Interrupts								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x0040			
Core to Host Push Button Interrupts								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x0044			
Watchdog Control Register																								0x0048				
Watchdog Timeout Register																								0x004C				
Watchdog Timer Value																								0x0050				
Watchdog Service Register																								0x0054				
Interrupt Mask Register																								0x0058				
Interrupt 0 Register																								0x005C				
Interrupt 1 Register																								0x0060				
Interrupt 2 Register																								0x0064				
Interrupt 3 Register																								0x0068				
PI00		PI09		PI02		PI03		PI04		PI05		PI06		PI07		0x006C												
PI08		PI09		PI10		PI11		PI12		PI13		PI14		PI15		0x0070												
PI16		PI17		PI18		PI19		PI20		PI21		PI22		PI023		0x0074												
PI24		TBD- if we need more interrupts																						0x0078				
ITAP Switch Test Mode																												

15.1.4.1 Switch Stage Numbers

31	27	26	24	23	16	15	11	10	8	7	0	Address Offset
0	0	0	0	ilink_mask00	ilink_spare00	0	0	0	0	0	stage num00	0x0000

16 iTAP Debug/Trace Interfaces

The iTAP Switch will have 2 different visibility points of the local Tensilica processor. The first visibility point is a trace port. The trace port allows one to monitor program flow by outputting information about the program counter. This is a non-intrusive interface that only monitors information. The second port is the JTAG debug port. This is a bi-directional communications pathway that allows a programmer to set breakpoints, single step through code, and peek/poke at processor state bits as well as memory locations.

In this document a brief overview of the debug interfaces on commercially available microprocessors is given, followed by the iTAP implementation of each.

16.1 Program Trace / Debug Support of other Processors

Most processors on the market today have some sort of JTAG on-chip debug support. This allows a programmer to interface to the chip and single step through his or her code. Breakpoints can be, memory modified and individual processor registers can be set or cleared. Fewer processors have a 'trace' port which allows the monitoring of a program. Often, software bugs only crop up when they have been running 'at speed' and the insertion of special debug code, or slowing down the system by single stepping through code mask the problem. Here is a brief list of what other processor vendors provide.

Motorola - Special 8 bit Debug trace port + on chip debug via JTAG

ARM - Embedded ICE. This provides JTAG control.]

Intel StrongArm - JTAG single step, etc.

SuperH - JTAG only

TI DSP - JTAG only

Analog Devices DSP - JTAG

MIPS - JTAG. (Some licensed vendors may support more, couldn't find any info).

Motorola is the only vendor that had debug module which provides for a program trace. This is used in their coldfire microprocessors. This is a byte wide interface which is divided into 2 nibbles. These two nibbles are the PST and DDATA.

PST[3:0]	Definition
0000	Continue Execution
0001	Begin Execution of an instruction
0010	Reserved
0011	Entry into user-mode
0100	Begin Execution of Pulse or WDATA
0101	Begin execution of a taken branch
0110	Reserved
0111	Begin execution of RTE inst.
1000	Begin 1 byte DDATA xfer
1001	Begin 2 byte DDATA xfer
1010	Begin 3 byte DDATA xfer
1011	Begin 4 byte DDATA xfer
1100	Exception Processing
1101	Emulator mode entry exception processing
1110	Processor is stopped, wait for irq
1111	Processor is halted

The shortfalls of this interface are that it uses an extra cycle to start sending program counter relative branch information and offers little insight as to why the processor has stalled. On the 'plus' side it has a special PULSE mode which can be set by the processor for performance mode. The

Proprietary and Confidential Information of Onex Communications Corporation

Tensilica debug interface gives extensive processor stall information. When it has more information to send than there is time to send it, the trace port will halt the processor. The iTAP debug interface borrows heavily from the Motorola family of processors, with some modifications.

16.2 iTAP Trace Port

This trace interface is used to follow a program trace of a Tensilica microprocessor. On the Tensilica processor, one of the configuration options is to create a Trace Port. This port gives a great deal of information regarding the processor's internal state and when it encounters pipeline bubbles, etc. Using it, one can determine the execution flow of a program in real time. Unfortunately, it is a 41 bit interface.

The iTAP Debug port is completely non-intrusive to the Tensilica core. It will never halt the processor. The debug port has the ability to allow tracing of the program counter, as well as load/store address or data words. The amount of data output is configurable by the programmer. It is output in real time at the internal core rate of the Tensilica processor.

Since the port is non-intrusive and has little buffering if too many jumps or load/stores occur in a row the data will be lost. Judicious use of which data to output will be necessary.

16.2.1 Tensilica Trace Port Interface

SUBJECT TO TENSILICA NDA

18 Clocking, Reset, and Initialization

18.1 Clocking

TBD

18.2 Initialization

Initialization of the master processor is described here, after its initialization is complete the switch datapath, high speed serial links and several other sub modules need to be initialized. Please consult their individual design specifications for correct sequence and initialization registers.

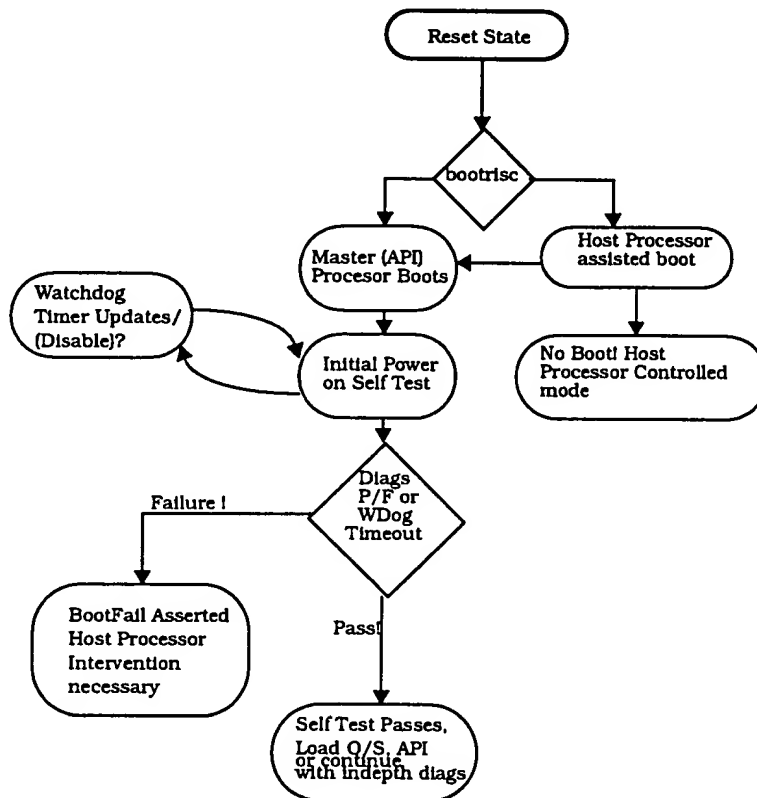


Figure 18-1: iTSE Initialization & Reset Sequence

- **Reset State**

Power has been applied to the iTSE, hresetx is asserted (active low). The BootDev, BootCfg, RiscClkSel and BootRisc pins are driven to the correct values. (See pin description for appropriate values). The choices allow for booting out of the SPI, Flash, FCRAM or Internal Ram. Hresetx is deasserted.

- **Master (API) Processor Boots**

Either the SPI has been selected OR the host has downloaded code to the external Host Expansion Ram / internal Ram and then wrote to the Master API Processor Reset bit. In either case the master processor now goes to its reset vector and begins executing code. This vector is fixed at processor generation time. Therefore several instructions will be hardcoded into the iTSE memory map at the processor's reset vector which perform a branch to the selected memory segment- ie flash, spi, host expansion ram or internal ram.

- **Initial Power on Self Test**

Coming out of reset the watchdog timer will be enabled with a default timeout of X ms. This will give the master processor enough time to execute several instructions and update the

watchdog timer.

- **Initial Power on Self Test Failure**

This state is reached by either of 2 conditions:

The master API processor never executes any good code due to either a severe chip manufacturing bug or a circuit board problem- open/shorts, etc. When this occurs the watchdog timer has never been cleared by software and expires.

The master API processor begins booting and during one of its tests discovers an error- for example during testing its Instruction RAM it detects a stuck bit- it can write to the BootFail bit and immediately assert it or it can let the watchdog expire.

- **Host Processor Assisted Boot**

The Host processor will access the internal memory or Host Expansion FCRam and download code to the API processor. It can then set the BootDev and BootCFG pins accordingly and set the RiscBoot pin.

- **No Boot**

No code is ever downloaded to the API processor, the iTSE is run in host-only mode.

19 I/O Definitions and Timing

20 Packaging and Pinout

20.1 Chiron Packaging

The following chart is taken from the IBM SA-27E databook.

1.27 mm pitch Laminate BGA	Max Signal I/O for Die-Package Combination				
	still looking for this info.				
42.5 x 42.5 mm: 1088 total balls		640	748	748	748
40 x 40 mm: 960 total balls		640	664	664	664
37.5 x 37.5 mm: 840 total balls		576	576	576	576
35 x 35 mm: 728 total balls		500	500	500	500
33 x 33 mm: 624 total balls	412	456	456	456	456
Metal layers	Wireable Gates (Millions) for Die-Size Metal Layer Combinations				
6lm	5.66	6.73	7.86	9.08	10.39

Table 20-1: SA-27E Organic Die-Package Menu: Single Density Footprint Laminate BGA

The iTap Switch is initially targeted at the 624 ball 1.27mm pitch BGA package.

A preliminary pin list for the iTap switch is given below:

20.2 Chiron I/O Summary

Functional Group	Signals	Number
Input Links	LINK_IN00_D00_P_N, LINK_IN00_D01_P_N, LINK_IN00_GNTOUT_P_N -12 of these-	72

Output Link	LINK_OUT00_D00_P_N, LINK_OUT00_D01_P_N, LINK_OUT00_GNTIN_P_N -12 of these-	72
Host Interface Address and Data	HADDR12-00, HDATA15-00	29
Host Interface Control	H_DSx,H_CSx,H_WEx1-0, H_DSACKx,H_IRQ1-0x	7
Host Expansion FCRAM	HEXP_ADDR (15),HEXP_DATA (16) ,HEXP_CNTRL (10),HEXP_VREF (2)	43
SPI Bus	SPICLK,SPIMOSI,SPIMISO,SPICS0,SPICS1	5
UART	TX,RX,D_TX,D_RX	4
Trace Port	TPSTAT[3:0],TPDATA[3:0],TPCLK	9
JTAG	TCK,TMS,TRSx,TDI,TDO	5
System Control	Ref Clock 1 (2), Ref Clock 2 (2), RiscClock (1) PLLA(4), PLLB(4), PLLC (4) HRESETx, SRESETx, SOR_SYNC (2), CTM[3:0]	25
Misc	BOOTRISC,BOOTDEV(2), BOOTCFG(2), RISC_CLK_SEL, Thermal(2), BOOTFAILx (1)	9
Test Access Port	TIB	64
Test Pins	TE,Z	2
Total I/O Pins Assigned		346
Total I/O Pins Unassigned		38
Total I/O Pins	Assigned I/O Pins + Unassigned I/O Pins	384
Dedicated Power (*1)	VCC (SSTL), VCC (Core), VCC(I/O), GND	168
Unilink Power	VCC & VSS for Unilink(1.8V), Unilink(2.5V),	72
Total Power Pins	Dedicate Power + Unilink Power	240
Total Package Pins	Total I/O Pins + Total Power Pins	624

Table 20-2: iTAP Switch Chip I/O

*1: These are dedicated power pins on the chip as defined by IBM document #SA14-2180-01 Lamineate Ball Grid Array p.97 Fig. 38

20.3 Signal Description

This sections describes the iTAP Switch pinout. Active low signals have an 'x' suffix.

20.3.1 Host Interface

Signal Name	Mnemonic	Direction	I/O Pad	Description
Address Bus	HADDR[12:00]	I	BT3335	Host Interface Address bus, word addressable, byte writable. It can address 16K bytes of internal memory
Data Bus	HDATA[15:00]	B	BT3335	Host Interface 16 bit data bus
Chip Select	HCSx	I	BT3335_PU	Host Interface Chip Select
Data Strobe	HDSx	I	BT3335	Host Interface Data Strobe
Data Acknowledge	DSACKx	O	BT3335_PU	Host Interface
Write Enable	HWE[1:0]	I	BT3335	Host Write Enable
Interrupts	HIRQ[1:0]	O	BT3335_PU	Host Interrupts

Total number of pins: 36

20.3.2 Host Expansion FCRAM Interface

Signal Name	Mnemonic	Direction		Description
-------------	----------	-----------	--	-------------

Reference Voltage	HEXP_VRef	O	VSSTL2R1	Voltage References (2)
Address Bus[14:00]	HEXP_ADDR	O	BSSTL2C2	Address Bus (15 bits)
Data Bus[15:00]	HEXP_DATA	B	BSSTL2C2	Data Bus (16 bits)
Function Select	HEXP_FN	O	BSSTL2C2	Address / Data Function Pin
Chip Select	HEXP_CSx	O	BSSTL2C2	Active low chip select
Bank Address	HEXP_BA[1:0]	O	BSSTL2C2	2 Bit Bank Address
CLK(+/-)	HEXP_CLK	O	BSSTL2C2	Differential Clock
Data Mask (Lower Byte)	HEXP_DML	O	BSSTL2C2	Write Mask (Data Bits 7-0)
Data Mask (Upper Byte)	HEXP_DMU	O	BSSTL2C2	Write Mask (Data Bits 15-8)
Lower Byte Data Strobe	HEXP_LDQS	B	BSSTL2C2	Data Strobe (Bits 7-0)
Upper Byte Data Strobe	HEXP_UDQS	B	BSSTL2C2	Data Strobe (Bits 15-8)

Total number of pins: 43

We have 2 Reference Voltage pins for 42 I/O. Is this enough?

20.3.3 SPI Bus

Signal Name	Mnemonic	Direction	I/O Pad	Description
SPI Clock	SPICLK	O	BT3335	SPI Clock.
SPI Master Out Slave In	SPIMOSI	O	BT3335	SPI Data Write
SPI Master In Slave Out	SPIMISO	I	BT3335_PD	SPI Data Read
SPI Chip Select	SPICSx[1:0]	O	BT3335	SPI Chip Selects

Total Number of pins: 5

20.3.4 UART

Signal Name	Mnemonic	Direction	I/O Pad	Description
Uart TX	TX	O	BT3335	Transmit
Uart RX	RX	I	BT3335_PD	Recive
Uart Daisy TX	DTX	I	BT3335_PD	Daisy Chained Transmit
Uart Daisy RX	DRX	O	BT3335	Daisy Chained Receive

Total Number of pins: 4

20.3.5 JTAG

Signal Name	Mnemonic	Direction	I/O Pad	Description
JTAG Clock	TCK	I	BT3335	JTAG Clock
JTAG Command	TMS	I	BT3335	JTAG Command
JTAG Reset	TRSx	I	BT3335	JTAG Reset (Active Low)
JTAG Data In	TDI	I	BT3335	JTAG Serial Data In
JTAG Data Out	TDO	O	BT3335	JTAG Serial Data out

Total number of pins: 5

20.3.6 Trace Port

Signal Name	Mnemonic	Direction	I/O Pad	Description
Trace Port Status	TPSTAT[3:0]	O	BT3335	Tensilica Program Counter Change Status
Trace Port Data	TPDATA[3:0]	O	BT3335	Tensilica Program Counter Relative Change

Trace Port Clock	TPCLK	O	BT3335	This is the core clock.
------------------	-------	---	--------	-------------------------

The speed of these pins = cclk, 250MHz nominal. An investigation is needed to determine what drivers should be used for these pins. Initially LVTTTL.

Total number of pins: 9

20.3.7 Miscellaneous

Signal Name	Mnemonic	Direction	I/O Pad	Description
Boot Risc Processor	BOOTRISC	I	BT3335_PD	Upon hreset, this bit is latched to determine if the risc core should boot. 0= No Boot 1= Boot.
Boot Device Select	BOOTDEV[1:0]	I	BT3335_PD	Upon reset, these bits are read to determine the method the processor should be bootstrapped 00 = Serial Prom (SPI) 01 = Internal Shared Ram 11 = External FCRAM
Boot Device Config	BOOTCFG[1:0]	I	BT3335_PD	Depending upon the boot device selected, these bits will take on different meanings. These are listed below:
				BOOTDEV=00=Serial Prom BOOTCFG[1] = Mem Size 0=16 bit addressing 1 = 24 bit addressing BOOTCFG[0] = SPI Device 0 = SPI CS 0 1 = SPI CS 1
				BOOTDEV=01=Internal Shared RAM BootCFG= unused
				BOOTDEV=11=FCRAM BOOTCFG = unused
Master Processor Clock Select	RISC_CLK_SEL	I	BT3335_PD	Upon reset, this bit is latched to determine the clock source of the embedded risc core. 0= Core Clock 1= Risc Clock
Power-On Boot Failure	BOOTFAILx	O	BT3335	Upon a Power On Boot Failure this pin will go active hi until reset
Thermal Diode	THERMALi	I	THERMAL	Thermal Diode on the iTAP Switch specified @ xxxxxxx
Thermal Diode	THERMALo	O	THERMAL	Thermal Diode on the iTAP Switch specified @ xxxxxxx

Total number of pins: 9

20.3.8 Switch Fabric Serial Links

Signal Name	Mnemonic	Direction	I/O Pad	Description
<i>Serial Data Links (Inbound Traffic)</i>				
Link_IN[11:00]_D[1:0]_P_N	LINK_IN00_D0_P	I	unilink pad name ???	Input Link Data (Unilink)
Link_IN[11:00]_GNT_P_N	LINK_IN00_GNT_P	O		Input Link Grant (Unilink)
<i>Serial Data Links (Outbound Traffic)</i>				

Link_OUT[11:00]_D[1:0]_P_N	LINK_OUT00_D0_P	O		Output Link Data (Unilink)
Link_OUT[11:00]_GNT_P_N	LINK_OUT00_GNT_P	I		Output Link Grant (Unilink)

Total number of pins: 144

20.3.9 System Control

Signal Name	Mnemonic	Direction		Description
Ref Clock 1 Input	PLLA_CLK(+/-)	I	IPECLD	PLL A Clock Input -primary input-**
Ref Clock 2 Input	PLLA_CLK(+/-)	I	IPECLD	PLL A Clock Input -primary input-**
RISC CLK	RISC_CLK	I	LVTTTL	Embedded Processor Clock
PLL A Feedback	PLLA_FB	O	BT3335	Look at ibm's pll data book spec.
PLL A Test Enable	PLLA_TST	I	BT3335PD	Test Input
PLL A Analog Power	PLLA_VDD	PWR		
PLL A Analog Ground	PLLA_GND	GND		
PLL B Feedback	PLLB_FB	O	BT3335	Look at ibm's pll data book spec.
PLL B Test Enable	PLLB_TST	I	BT3335PD	Test Input
PLL B Analog Power	PLLB_VDD	PWR		
PLL B Analog Ground	PLLB_GND	GND		
PLL C Feedback	PLLC_FB	O	BT3335	Look at ibm's pll data book spec.
PLL C Test Enable	PLLC_TST	I	BT3335PD	Test Input
PLL C Analog Power	PLLC_VDD	PWR		
PLL C Analog Ground	PLLC_GND	GND		
Clock Test Mode Pins	CTM[3:0]	O	BT3335	Clock Test Pins
Hardware Reset	HRESETx	I	BT3335	Hardware Reset - resets everything in the chip
Software Reset	SRESETx	I	BT3335	Software Reset- controlled reset, processor, only certain registers are reset
Start of Row	SOR_SYNC(+/-)	I	IPECLD	System-wide Start of Row Pulse

Total Number of pins: 25

20.3.10 Test Access Port

Signal Name	Mnemonic	Direction	I/O Pad	Description
General Purpose Test	TIB[63:00]	I/O	BT3335PDT	General Purpose Test I/Os They will also be the dedicated IBM test pins.
Test Enable	TE	I	IT33TEPDT	IBM LSSD Dedicated Test Enable Pin
High Z	Z	I	BT3335	Tie hi to tri-state all I/Os for ATPG Testing

Total Number of pins: 66

20.3.11 Power Supplies

Signal Name	Mnemonic	# of Pins
VCC (Core - 1.8V)	DVCC18	48
VCC (3.3V)	DVCC33	40
VSS (3.3) & (1.8)	DVSS	80
UniLink VCC18	LVCC18	18

UniLink VCC25	LVCC25	18
UniLink VSS18	LVSS18	18
UniLink VSS25	LVSS25	18

20.3.12 Reference Documents

IBM SA14-2180-01 Laminate Ball Grid Array

IBM SA14-2208-02 ASIC SA-27E Databook

20.4 Package Outline

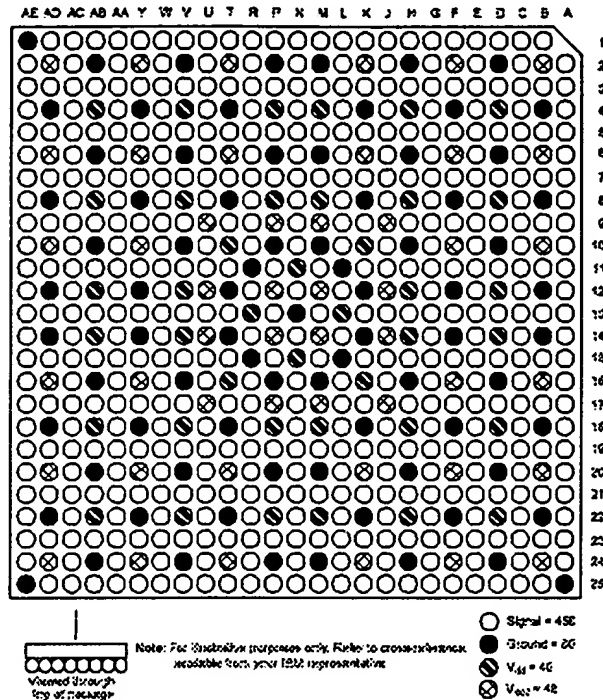


Figure 20-3: Laminate BGA 624 ball, 1.27 mm pitch, dual/single power zone

21 Electrical Characteristics

RESERVED

21.1 I/O Drivers

RESERVED

21.1.1 UniLink

RESERVED

21.1.2 LVTTTL (5V Tolerant I/O)

RESERVED

21.1.3 LVTTTL

RESERVED

